

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
Ingeniero de Sistemas

TEMA:
**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA DISPOSITIVOS
ANDROID, PARA LA RESERVA DE LABORATORIOS, MATERIALES Y
LA GESTIÓN DE NOTIFICACIONES DE MANTENIMIENTO DE EQUIPOS
DEL CECASIS**


Autor:
FABRICIO RICARDO MALDONADO TAIPE

TUTOR:
GUSTAVO ERNESTO NAVAS RUILOVA

Quito, agosto del 2018

CESIÓN DE DERECHOS DE AUTOR

Yo, FABRICIO RICARDO MALDONADO TAIPE, con documento de identificación N° 1725269151, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación con el tema: Desarrollo de una aplicación móvil para dispositivos Android, para la reserva de laboratorios, materiales y la gestión de notificaciones de mantenimiento de equipos del CECASIS, mismo que ha sido desarrollado para optar por el título de INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



FABRICIO RICARDO
MALDONADO TAIPE
CI: 1725269151

Quito, agosto del 2018

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: Desarrollo de una aplicación móvil para dispositivos Android, para la reserva de laboratorios, materiales y la gestión de notificaciones de mantenimiento de equipos del CECASIS realizado por el estudiante FABRICIO RICARDO MALDONADO TAIPE, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto del 2018

A handwritten signature in blue ink, reading "Gustavo Ernesto Navas R.", with a dotted line underneath.

GUSTAVO ERNESTO NAVAS RUILOVA

CI: 1705675625

DEDICATORIA

A mis padres, los cuales me han enseñado con su gran sabiduría a nunca rendirme, con su gran paciencia a ser mejor ser humano, con su gran carisma a sonreír toda mi vida, con su gran ejemplo a superarme cada día más y en especial con su gran amor que lo demuestran cada día de su vida.

A mi hermana ejemplar que me motiva a superarme mis sueños, que me obliga a perder los miedos y cumplir mis metas. “La extraño muchísimo”.

A todos mis tíos y mis tías, mis primos y mis primas que siempre están pendiente de nuestro bienestar y nuestro existir.

A Valeria que siempre me apoyo en las buenas y en las malas, una gran persona que fue de gran ayuda en todo el proceso de titulación.

Y en especial a Dios el cual me escucha y me apoya cada día de mi vida.

AGRADECIMIENTO

Yo agradezco a la Universidad Politécnica Salesiana que ha contribuido en nuestra formación profesional, al grupo del CECASIS el cual no ha brindado las facilidades necesarias para realizar el proyecto de titulación.

Fabricio Ricardo Maldonado Taipe.

ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	2
DECLARATORIA DE COAUTORÍA DEL TUTOR.....	3
DEDICATORIA	4
INDICE DE FIGURA S.....	10
RESUMEN.....	12
ABSTRACT.....	13
Introducción	1
Antecedentes	1
Problema	2
Justificación.....	2
Objetivo general.....	3
Objetivos específicos	3
Alcance del proyecto.....	3
Capítulo 1	6
1 Marco teórico.	6
1.2. Herramientas de desarrollo.	6
1.2.1. Plataformas de desarrollo.	6
1.2.3. Base de datos.....	12
1.2.4. Dependencias y librerías.	12
1.2.5. Servicio web.....	15
1.2.6. Marco Metodologico.....	15
Capítulo 2.....	16
2. Análisis y diseño.	16
2.1. Requerimientos iniciales.	16
2.1.1. Requerimientos de hardware.....	16
2.1.2. Especificación de requerimientos de software.	17
2.2. Requerimientos específicos.....	19
2.2.1. Procesos as is.	19
2.2.2. Procesos to be.....	21
2.3. Requerimientos no funcionales.	24
2.4. Requerimientos funcionales.....	25
2.4.1. Diagramas de casos de uso.....	25
2.5. Diseño de la base de datos.	35
2.5.1. Diseño de base de datos de autenticación e ingreso.....	36

2.5.2. Diseño de base de datos de reserva de laboratorio.....	36
2.5.3. Diseño de base de datos de disponibilidad de materiales.	37
2.5.4. Diseño de base de datos de gestion de mantenimiento (incidentes).	38
2.6. Diseño de autenticación de aplicación mediante tokens.	39
2.6.1. Diseño del servicio web.	40
2.7. Diseño de clases.	41
2.7.1. Diagrama de clases.....	42
2.8. Diagrama de despliegue.	46
Capítulo 3.....	47
3. Construcción.	47
3.1. Construcción de la aplicación móvil.	47
3.1.1. Arquitectura de la aplicación.	47
3.2. Construcción de interfaz y funcionalidad.	48
3.2.1. Construcción de la autenticación e ingreso.....	48
3.2.2. Construcción de noticias.	50
3.2.3. Construcción de módulo laboratorio.....	52
3.2.4. Construcción de módulo materiales.	55
3.2.5. Construcción de módulo mantenimiento.	56
3.3. Construcción de servicios web.....	58
3.3.1. Servicio web de autenticación e ingreso.	58
3.3.2. Servicio web de consulta de laboratorios.....	59
3.3.3. Servicio web de consulta de materiales.	60
3.3.4. Servicio web de consulta de incidentes.....	60
4. Pruebas del software	62
4.1. Software y hardware.	62
4.2. Prueba de rendimiento de la base de datos.	63
4.3. Pruebas de aceptación.	65
4.3.1. Perfil de usuarios.....	66
Conclusiones	74
Recomendaciones.....	75
Bibliografía	76
Anexos	¡Error! Marcador no definido.
Anexo 1: prototipo	¡Error! Marcador no definido.
Prototipo reserva de laboratorios	¡Error! Marcador no definido.
Prototipo consulta de disponibilidad de materiales.....	¡Error! Marcador no definido.

Anexo 2: carta de aceptacion del software.....**¡Error! Marcador no definido.**

ÍNDICE DE TABLAS

Tabla 1. Caso de Uso Autenticación e Ingreso	25
Tabla 2. Caso de uso Reserva de Laboratorios	27
Tabla 3. diagrama de caso de uso: consulta de materiales	29
Tabla 4. Diagrama de Casos de Uso: Gestión de Mantenimiento.....	32
Tabla 5. Diagrama de caos de Uso: Noticias	33
Tabla 6. Valores de la prueba de carga y rendimiento de la BDD.....	64
Tabla 7. Perfiles de Usuario	66
Tabla 8. Prueba de aceptación: login	67
Tabla 9. Pruebas de Aceptación; Visualizar Noticias	68
Tabla 10. Pruebas de Aceptacion: Ver detalles de Noticias.....	69
Tabla 11. Pruebas de Aceptación: Login recordando Contraseña	69
Tabla 12. Prueba de aceptación: Buscar y Visualizar Laboratorio disponibles	70
Tabla 13. Prueba de Aceptación: Visualizar los materiales Disponibles	71
Tabla 14. Prueba de Aceptación: Ingresar Incidencia.....	72
Tabla 15. Pruebas de Aceptación: Ver incidencias	72
Tabla 16. Prueba de Aceptación: Ver Asignación de Incidencias	73

ÍNDICE DE FIGURAS

Figura 1. PBS de la aplicación Móvil.	5
Figura 2. Ejemplo de componentes de BPMN.....	10
Figura 3. PBS que modela los componentes del proyecto.	18
Figura 4. Ilustración. Procesos AS IS Reserva de laboratorios.	19
Figura 5. Procesos AS IS Consulta de Materiales.....	20
Figura 6. Procesos AS IS informe de Mantenimiento.....	21
Figura 7. Procesos To Be Consulta y Reserva de Laboratorios.....	22
Figura 8. Proceso TO BE de Consulta de Materiales.....	23
Figura 9. Procesos TO BE de Gestión de Mantenimiento.	23
Figura 10. Diagrama de casos de uso: Autenticación e Ingreso.	26
Figura 11. Diagrama de casos de Uso: Reserva de Laboratorio.	28
Figura 12. Diagrama de casos de uso: Consulta de Materiales.....	30
Figura 13. Diagrama de Casos de Uso: Consulta de Materiales.....	32
Figura 14. Diagrama de Caso de Uso: Consulta de Materiales.....	34
Figura 20. Base de Datos del Sistema Web.	35
Figura 21. Tablas de Autenticación e Ingreso.....	36
Figura 22. Tablas de Reserva de Laboratorios.....	37
Figura 23. Tablas de Disponibilidad de Materiales.	38
Figura 24. Tablas para Gestión de Mantenimiento.	39
Figura 25. Sistema de autenticación entre aplicaciones.....	40
Figura 26. Código para generar un token único.....	41
Figura 27. Código de comparación entre el token asignado y el token del servidor de aplicaciones.....	41
Figura 28. Diagrama de Clases Autenticación e Ingreso.	42
Figura 29. Diagrama de Clases de Reserva de laboratorios.....	43
Figura 30. Diagrama de clases de Materiales.....	44
Figura 31. Diagrama de Clases de Gestión de Mantenimiento.....	45
Figura 32. Diagrama de despliegue de la aplicación web y móvil.....	46
Figura 33. Diagrama: Arquitectura de la APP Móvil.	47
Figura 34. Componentes utilizados en Autenticación e Ingreso.....	48
Figura 35. Interfaz de Autenticación e Ingreso.....	49
Figura 36. Código de Inicio de Sesión.....	49
Figura 37. Código Guardar estado de Inicio de Sesión.....	50
Figura 38. Código Guardado en variables de memoria interna.	50
Figura 39. Código Guardado en variables de memoria interna.	50
Figura 40. Componentes utilizados en Autenticación e Ingreso.....	51
Figura 41. Interfaz de Noticias.....	51
Figura 42. Llamada Asíncrona a el servicio web noticias.....	52
Figura 43. Llamada Asíncrona a las imágenes en el servidor web.	52
Figura 44. Interfaz de Reserva de Laboratorios.....	53
Figura 45. Llamada a DatePicker de Android.....	53
Figura 46. Interfaz de ver laboratorios.....	54
Figura 47. Componentes utilizados en ver Laboratorios.	54
Figura 48. Código Declaración de los Arrays para exponer en el ListView.....	55
Figura 49. Interfaz de Consulta de Materiales.	55

Figura 50. Componentes utilizados en ver Laboratorios.	56
Figura 51. Código búsqueda de Materiales desde un ListView.....	56
Figura 52. Interfaz de Ingreso de Incidente.	57
Figura 53. Componentes utilizados en el ingreso de incidentes.	57
Figura 54. Código de AlertDialog para guardar el incidente.	58
Figura 55. Datos enviados y respuestas json de Autenticación e ingreso.....	59
Figura 56. Datos enviados y respuestas json de consulta de laboratorios.....	59
Figura 57. Datos enviados y respuestas json de consulta de materiales.	60
Figura 58. Datos enviados y respuestas json de consulta de incidentes.....	61
Figura 59. Puntos críticos de la prueba de carga y rendimiento.	65
Figura 60. Prototipo de Noticias.	¡Error! Marcador no definido.
Figura 61. Prototipo de popUp noticias.	¡Error! Marcador no definido.
Figura 62. Prototipo de reserva de laboratorios.	¡Error! Marcador no definido.
Figura 63. Prototipo de filtrado de laboratorio por software.	¡Error! Marcador no definido.
Figura 64. Prototipo de popup de software.	¡Error! Marcador no definido.
Figura 65. Prototipo de reserva de laboratorios.	¡Error! Marcador no definido.
Figura 66. Prototipo de Ver laboratorios.	¡Error! Marcador no definido.
Figura 67. Prototipo de Inicio de materiales.	¡Error! Marcador no definido.
Figura 68. Prototipo de búsqueda de materiales.	¡Error! Marcador no definido.
Figura 69. Carta de aceptación del software.	¡Error! Marcador no definido.

RESUMEN

Existe una pérdida de recursos por parte del personal de CECASIS ya que lleva un registro manual por lo cual el presente proyecto está orientado al desarrollo de una aplicación móvil, la cual será implementada en la plataforma Android, dedicada a los estudiantes, profesores y personal de CECASIS la cual facilite la reserva de laboratorios, consulta de materiales y reporte de incidentes.

Para el desarrollo de esta aplicación se utilizó la metodología Scrum, ya que es una metodología ágil que permite llevar el control de cada avance y entregable durante el desarrollo del software.

Como resultado se encontró que la usabilidad de la aplicación gracias a las normas de desarrollo de dispositivos móviles logró un mejor entendimiento y aprendizaje de la misma, permitiendo un mejor análisis de datos y manejo de recursos con la información digitalizada que genera el registro del sistema.

ABSTRACT

The CECASIS staff undergoes resources waste as it has a manual registration system, this is the reason why this project aimed to develop a mobile application that will be implemented on the Android platform; whose users will be students, teachers and CECASIS staff to help booking the laboratories, material consultation and incidents' reports.

This application was developed with Scrum methodology, since it is an agile methodology that allows to take control of each progress made and their deliverables during the software development.

The results showed, due to the mobile devices development rules, that the usability of the application allows a better understanding and learning of it, besides, a better data analysis and resources management with the digitized information generated by the system registry.

Introducción

Antecedentes.

Debido al avance de la tecnología, en especial de la tecnología móvil, cada día las personas desean aumentar la agilidad en los procesos y reducir tiempos muertos que producen inconformidad; por esto, la mayoría de las empresas alrededor del mundo, privadas y públicas, crean aplicaciones móviles para mejorar la experiencia del usuario final y lograr una mayor satisfacción del cliente.

El centro de capacitaciones y servicios informáticos de la Universidad Politécnica Salesiana Campus Sur (CECASIS) ofrece soporte preventivo y correctivo a las computadoras situadas en los laboratorios.

Servicios del CECASIS.

- Servicios de préstamo de hardware para estudiantes internos de la Universidad Politécnica Salesiana.
- Servicio de capacitación para estudiantes internos y externos a la Universidad Politécnica Salesiana.
- Mantenimiento preventivo como correctivo de las computadoras ubicadas los laboratorios de la Universidad Politécnica Salesiana.
- Soporte de incidentes y problemas a las computadoras de los laboratorios ubicadas en el CECASIS.

Problema.

Los estudiantes y profesores de la Universidad Politécnica Salesiana notan la existencia de una pérdida de tiempo de los estudiantes y profesores al momento de realizar la reserva de un laboratorio o la petición de un material en el CECASIS ya que se necesita estar personalmente para realizar esta tarea. Con una aplicación móvil esto sería mucho más sencillo, eficiente, rápido y fácil de gestionar.

Existe una pérdida de recursos por parte del personal de CECASIS ya que lleva un registro manual lo cual dificulta la gestión de laboratorios, el análisis y la toma de decisión en base a esa información registrada.

Justificación.

Ofrecer a los alumnos de la UNIVERSIDAD POLITÉCNICA SALESIANA una aplicación móvil que mejore la eficiencia en el CECASIS en la gestión de laboratorios, materiales y mantenimiento apoyando a los objetivos estratégicos de la Universidad, al mejorar el acceso a la información relevante, concreta y oportuna a través de las tecnologías de la información y comunicación, como consecuencia reducirían costos lo que apoyará una gestión económica financiera, que facilita la toma de decisiones y el cumplimiento de los objetivos institucionales y ayudará a la comunidad universitaria de la UPS contar con información gestionada y conservada eficientemente.

Objetivo general.

Desarrollar una aplicación móvil para dispositivos Android, para la reserva de laboratorios, consulta los materiales disponibles y gestiona las notificaciones de mantenimiento de equipos del CECASIS y de este modo aportar el incremento de la eficiencia del grupo de trabajo.

Objetivos específicos.

- Levantar y diagramar procesos con la notación BPMN para tener una visión más clara del funcionamiento del CECASIS.
- Especificar los requerimientos necesarios con un modelo interactivo para lograr desarrollar una aplicación que cumpla con las necesidades de los interesados.
- Diseñar una aplicación móvil en Android Studio que sea fácil de aprender y usar.
- Codificar la aplicación tomando en cuenta la limitación del Hardware de los teléfonos móviles.
- Utilizar el patrón MVC en la implementación de la aplicación para que el mantenimiento y futuras versiones sea más fácil de realizar.
- Realizar pruebas de validación de las funcionalidades requeridas.

Alcance del proyecto.

Para definir el alcance del proyecto se hizo uso de la Estructura de Descomposición del Producto o Product Breakdown Structure (PBS), en donde podemos presentar las partes funcionales de la APP.

La App móvil contará con:

- Ingreso y Autenticación con un registro previo de la aplicación móvil en el sistema web para la autenticidad de aplicaciones y un ingreso asignando un token para el consumo de los servicios web.
- Noticias CECASIS.
Todos los usuarios tendrán acceso a ver las noticias y el detalle de las noticias del CECASIS.
- Reserva de Materiales.
Este módulo gestionará la reserva de laboratorios, para los estudiantes y los docentes de la Universidad Politécnica Salesiana.
- Consultas de Materiales.
Todos los usuarios tendrán acceso a ver la disponibilidad de los materiales para préstamos en el CECASIS.
- Reporte de Incidentes.
Todos los usuarios tendrán acceso a reportar un incidente de maquina o laboratorio.

PBS del proyecto.

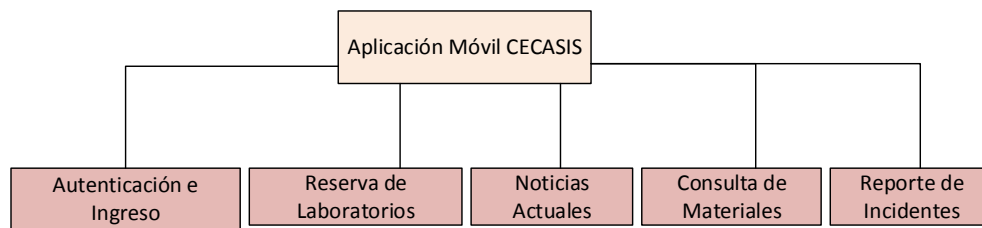


Figura 1. PBS de la aplicación Móvil.

Realizado por Fabricio Maldonado.

Capítulo 1

1 Marco teórico.

A continuación, se presenta las herramientas y la metodología de desarrollo de la aplicación móvil.

1.2. Herramientas de desarrollo.

1.2.1. Plataformas de desarrollo.

1.2.1.1. Android Studio.

Android Studio es un IDE (Entorno de desarrollo Integrado) para la plataforma Android basado en el software IntelliJ IDEA especialmente diseñado para el desarrollo en Android.

Características:

- Renderizado en tiempo real.
- Construcción con Gradle.
- Soporte para desarrollar en Android Wear.
- Plantillas para el diseño de interfaces.
- ProGuard, firma de aplicaciones.
- Edición gráfica de componentes.
- Emulador Android para ejecutar y probar aplicaciones.

(developer.android.com, 2017).

1.2.2.2. Bizagi modeler.

Bizagi es un Freeware que es utilizado para diagramar, documentar y simular procesos realizados con la notación BPMN (Business Process Model and

Notation). Permite ejecutar flujos de trabajo y es considerada como una solución a la gestión de procesos. Bizagui automatiza procesos complejos.

(Bizagi, 2018).

1.2.2.3. Sap powerdesigner.

PowerDesigner es una herramienta de modelado empresarial colaborativo que tiene soporte para diferentes tipos de modelos. El más importante para este proyecto es el Modelo de datos el cual funciona con la mayoría de sistemas de base de datos relacionales y utilizar para crear el modelo conceptual y físico de la base de datos.

PowerDesigner permite visualizar, analizar y manipular metadatos para tener una mejor arquitectura empresarial de información brindando un enfoque basado en modelos lo cual permite a la tecnología alinear con el giro del negocio logrando una mejor implementación de arquitecturas de información.

Está diseñado con buenas prácticas de análisis, diseño y gestión de metadatos brindando una mejor productividad individual y grupal.

(Wang, 2016).

1.2.2.4. Servidor http apache.

HTTP Apache es un servidor web multiplataforma que implementa el protocolo HTTP, es altamente configurable ya que es modular y extensible. Este es un programa diseñado para transferir datos de hipertexto (páginas web) con todos sus elementos.

Apache es el componente de la plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación Python/Perl/ PHP y Ruby, ya que contiene varios paquetes redistribuidos de software propietario.

La licencia de Apache permite distribuir la derivación de código abierto y cerrado a partir de su código fuente original.

(Apache Software Foundation., 2018).

1.2.2.5. Xampp.

Xampp es un software libre que gestiona el servidor apache, el gestor de base de datos MySQL, contiene intérpretes para lenguajes de Script PHP y Pearl.

Permite la instalación sencilla de Apache en el ordenador, sin importar el sistema operativo y es una herramienta que permite probar tu trabajo en tu propio ordenador sin necesidad de tener acceso a internet.

Paquetes importantes que se instalan con XAMPP:

- Apache, el servidor Web más famoso.
- MySQL, una excelente base de datos de código libre.
- PHP y Perl: lenguajes de programación.
- ProFTPD: un servidor FTP.
- OpenSSL: para soporte a la capa de sockets segura.

(Apache Friends, 2018).

1.2.2.6. Java.

Java es un lenguaje de programación orientado a objetos, actualmente es un lenguaje más popular alrededor del mundo ya que tiene la capacidad de escribir

una vez el código que se ejecutaría en cualquier dispositivo gracias a que cuenta con JVM o Java Virtual Machine.

Ventajas de utilizar Java:

- Lenguaje simple.
- Lenguaje orientado a objetos.
- Aplicaciones distribuidas.
- Interpretado y compilado.
- Lenguaje Seguro.

(Oracle, 2018).

1.2.2.7. Php.

PHP es un lenguaje de código abierto utilizado especialmente para el desarrollo web el cual puede ser incrustado en HTML.

PHP puede emplearse en todos los sistemas operativos también puede ejecutarse en cualquier servidor web que pueda utilizar el binario de PHP.

(The PHP Group, 2018).

1.2.2.8. Json.

JSON (Notación de Objetos de JavaScript) es un formato liviano de intercambio de datos utilizado mayormente para la transmisión de información de servicios web.

JSON está formado por dos estructuras:

- Pares de nombre y valor.

- Una lista ordenada de valores, se los conocen como arreglos, vectores, listas, etc.

(Nederlands, 2018).

1.2.2.9. Bpmn.

BPMN (Modelo y Notación de Procesos de Negocio) es una notación estandarizada que permite modelar procesos de negocios en forma de flujo de trabajo.

Ejemplo de diagrama de proceso de negocio.

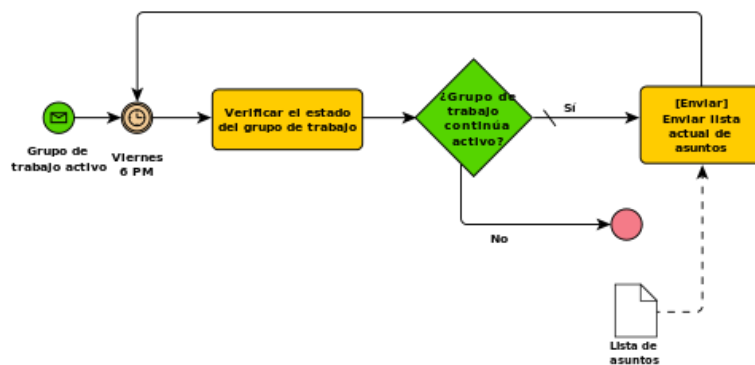


Figura2. Ejemplo de componentes de BPMN.

Realizado: (Debevoise, 2016).

La principal misión de esta notación es facilitar de manera gráfica un estándar de procesos que sea fácil legible y legible.

(Debevoise, 2016).

1.2.2.10. Gradle.

Gradle es un sistema de automatización de construcción de código que introduce un lenguaje específico de dominio (DSL) basado en Groovy.

Ventajas de Gradle en Android:

- Reutilización de código.
- Fácil configuración y personalización en la compilación.
- Gestiona las dependencias de forma potente y cómoda.
- Fácil creación de diferentes versiones de la aplicación.

(Gradle Inc, 2018).

1.2.2.11. Pencil project.

Pencil Project es una herramienta de creación de prototipos funcionales de código abierto que está disponible para todos los sistemas operativos.

(Evolus, 2018).

1.2.2.12. Material designer.

Material Design es una guía para el diseño visual, de movimientos y de interacción en Android brindando los siguientes elementos que permitirán crear aplicaciones Material Designer:

- Un tema nuevo Material Designer.
- Nuevos widgets para vistas complejas.
- Nuevas API (interfaces de programación de aplicaciones) para sombras y animaciones personalizadas.

(Google, 2018).

1.2.2.13. Apache jmeter.

La aplicación Apache JMeter es un software de código abierto diseñado para realizar pruebas de carga y rendimiento funcional de una aplicación.

Se puede usar para simular una gran carga en un servidor para probar su rendimiento general por medio de diferentes tipos de carga.

(Apache Software Foundation, 2018).

1.2.3. Base de datos.

A continuación, se presenta la base de datos utilizada de la aplicación móvil.

1.2.3.1. Postgresql.

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos.

Características.

- Alta concurrencia.
- Amplia variedad de tipo de datos.
- PostgreSQL provee nativamente soporte para:
 - Números de precisión arbitraria.
 - Texto de largo indefinido.
 - Arrays.
 - Diferentes complementos para geo análisis.

(Riggs, 2018).

1.2.4. Dependencias y librerías.

A continuación, se presenta las dependencias y librerías utilizadas en la aplicación móvil.

1.2.4.1. Pdo.

La Extensión Objetos de Datos es una interfaz para poder acceder a diferentes bases de datos en PHP exponiendo características concretas de la base de datos a conectar.

Para lograr una conexión se debe utilizar un controlador de PDO específico de la base de datos para tener acceso a un servidor de bases de datos que se lo habilita en PHP.

(The PHP Group, 2018).

1.2.4.2. Json_encode.

Un uso habitual de JSON es leer datos de un servidor web y exponer los datos en una página web.

Json_encode: Devuelve la información JSON del valor dado.

(Group, s.f.).

1.2.4.3. Query.

Query se utiliza cuando se quiere crear una consulta a una base de datos relacional y no relacional. Es una consulta SQL.

1.2.4.4. Biblioteca appcompat.

Esta biblioteca añade compatibilidad con los componentes de la interfaz de usuario de material design.

Appcompat v7 incluyen algunas clases como:

- **ActionBar:** proporciona una implementación de la barra de acciones.
- **AppCompatActivity:** añade una clase base para diálogos temáticos de AppCompatActivity.
- **ShareActionProvider:** añade compatibilidad para un uso compartido como correo electrónico o publicación en aplicaciones sociales.

(Google, 2018).

1.2.4.5. Support:design.

La biblioteca de soporte añade soporte para varios componentes de diseño, como cajones de navegación, botones de acción flotante, barras de aperitivos y pestañas.

(Google, 2018).

1.2.4.6. Android-async-http.

Android-Async-http es un cliente Http basado en la devolución de llamadas asincrónicas en Android.

Todas las solicitudes se realizan fuera de la secuencia de interfaz de usuario principal de su aplicación, pero cualquiera devolución de llamada se ejecutará en el mismo subproceso que se creó mediante el uso de mensajes del Manejador de Android.

(Smith, Android Asynchronous Http Client, 2016).

1.2.4.7. Smartimageview.

SmartImageView es una dependencia que permite cargar imágenes de un URL y ser almacenadas en cache, en memoria y en el disco para una carga súper rápida.

Características:

- La carga Asincrónica ocurre fuera del hilo de la interfaz de usuario.
- Las imágenes se acumulan en caché en la memoria y en el disco para una carga súper rápida.
- SmartImage puede cargar desde otras fuentes.

(Smith, Android Smart Image View, 2016).

1.2.5. Servicio web.

A continuación, se presenta los servicios web utilizados en la aplicación Android.

1.2.5.1. Servicios rest.

Un servicio web es un grupo de protocolos utilizados para intercambiar datos entre aplicaciones.

Los softwares creados en diferentes lenguajes de programación pueden usar servicios web para intercambiar datos.

- GET: Proporciona acceso de solo lectura a datos.
- POST: Se usa para modificar o crear datos.

(World Wide Web, 2016).

1.2.6. Marco Metodológico.

A continuación, se presenta la metodología utilizada en la realización de la aplicación móvil.

1.2.6.1. Scrum.

Scrum es una metodología de desarrollo ágil para software por lo cual es incremental interactiva. Se utiliza para guiar las siguientes actividades: requerimientos, análisis, diseño evolución y entrega.

1.2.6.1. Spring.

Un sprint es la unidad de tiempo que determina un ciclo de desarrollo con Scrum.

1.2.6.2. Prototipado.

El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

El diseño del prototipo se centra en aquellos aspectos del software que serán visibles para el usuario final, el cual es evaluado por el cliente para una retroalimentación así refinamos los requisitos del software que se desarrollará.

El prototipado ayuda al desarrollado y al cliente a entender de mejor manera cuál será el resultado final del software y así lograr la satisfacción del cliente. De esta manera se involucra al cliente para lograr mejor agilidad en la construcción de la aplicación.

(Maner, 2018).

Capítulo 2

2. Análisis y diseño.

En este capítulo se presenta el análisis previo realizado para cumplir con los objetivos planeados, así como los casos de uso y el prototipo a utilizarse para la construcción de la aplicación móvil.

2.1. Requerimientos iniciales.

A continuación, se presenta los requerimientos iniciales de la aplicación móvil.

2.1.1. Requerimientos de hardware.

Los requerimientos de hardware utilizados para el desarrollo de la aplicación en Android Studio son:

2.1.1.1. Computador.

- Intel Core i7 6500U de 2.6 GHZ.
- Memoria Ram: 12GB.

- Tarjeta de video: AMD Radeon R5 m335 de 4GB.
- Disco Duro: 1TB.

2.1.1.2. Dispositivo móvil.

- Marca: Nokia 6.
- Pantalla: 5.5 pulgadas.
- Resolución: 1,080x1,920 pixeles.
- Procesador: Snapdragon 430 de ocho núcleos (1.4GHz).
- RAM: 3GB.
- Almacenamiento: 32GB.
- Ranura para tarjeta microSD: Sí.
- Batería: 3,000mAh.
- Cámara trasera: 16 megapíxeles.
- Cámara frontal: 8 megapíxeles.
- Sistema operativo: Android 8 Oreo.
- Conectividad: Bluetooth 4.1, Wi-Fi 802.11n (2.4GHz y 5GHz).

2.1.2. Especificación de requerimientos de software.

2.1.2.1 Propósito.

El presente proyecto tiene como propósito facilitar una aplicación móvil que automatice la gestión de reserva de laboratorios, disponibilidad de materiales y gestión de incidentes técnicos.

La herramienta debe funcionar conjuntamente con la aplicación web desarrollada en el CECASIS.

La aplicación móvil debe permitir al profesor reservar un laboratorio exitosamente.

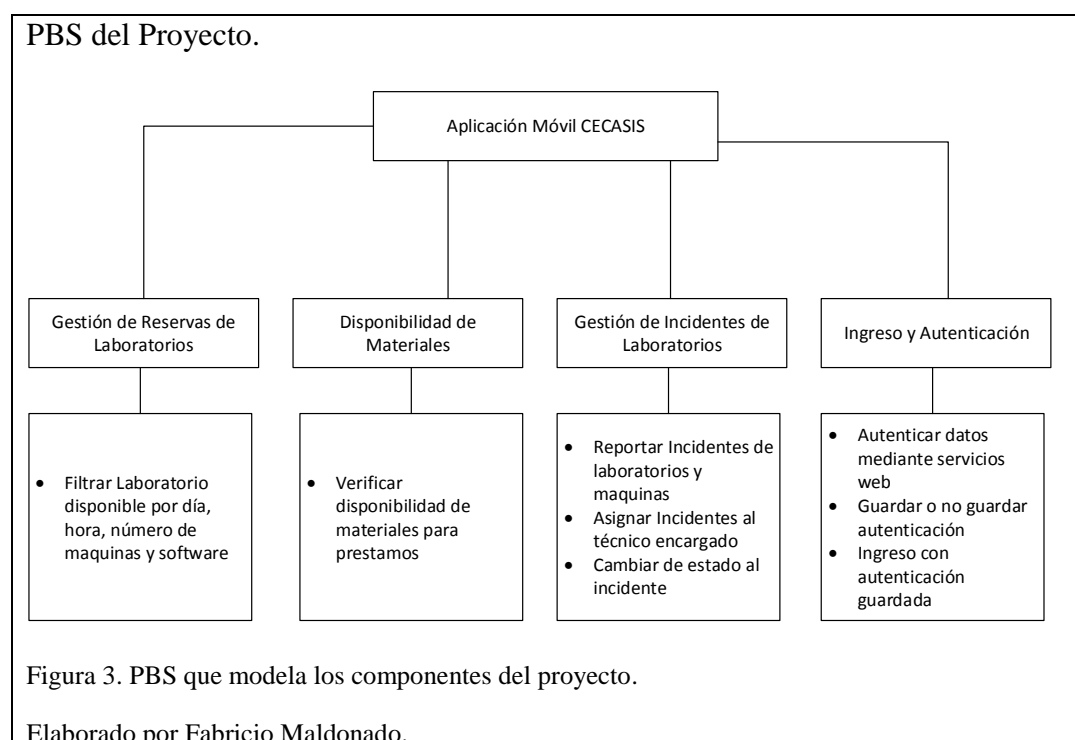
La aplicación móvil debe gestionar el mantenimiento de los laboratorios exitosamente.

La aplicación móvil debe verificar la disponibilidad y existencias de materiales exitosamente.

La aplicación móvil debe visualizar las noticias del CECASIS exitosamente de manera agradable a la vista.

2.1.2.2. Alcance.

El proyecto se divide en cinco etapas: ingreso y autenticación con servicios web, revisión de ultimas noticias del CECASIS, disponibilidad y reserva de laboratorios según autenticación, disponibilidad de materiales que ofrece CECASIS, gestión de incidentes de laboratorios y materiales, cierre de sesión.



2.2. Requerimientos específicos.

A continuación, se presenta los procesos funcionales del CECASIS:

2.2.1. Procesos as is.

Se utilizó los Proceso AS IS o “tal como está” que es levantamiento de procesos para tener una perspectiva de que se debe automatizar.

2.2.1.1. Reserva de laboratorios.

La reserva de laboratorios se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

Procesos AS IS Reserva de Laboratorios.

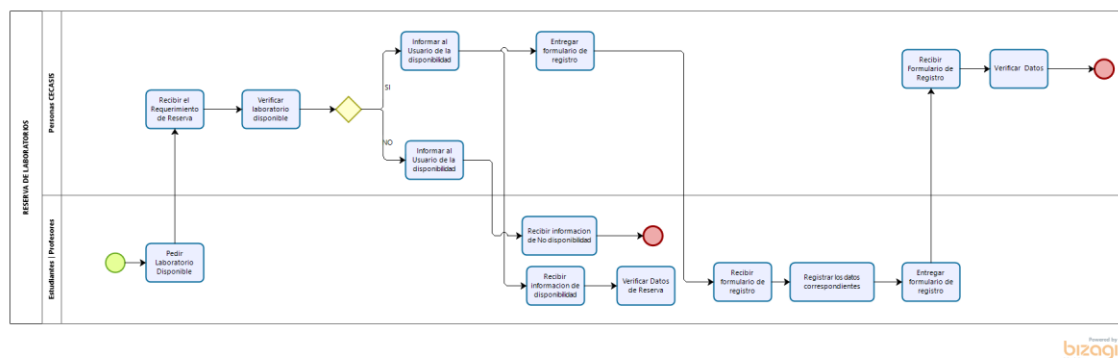


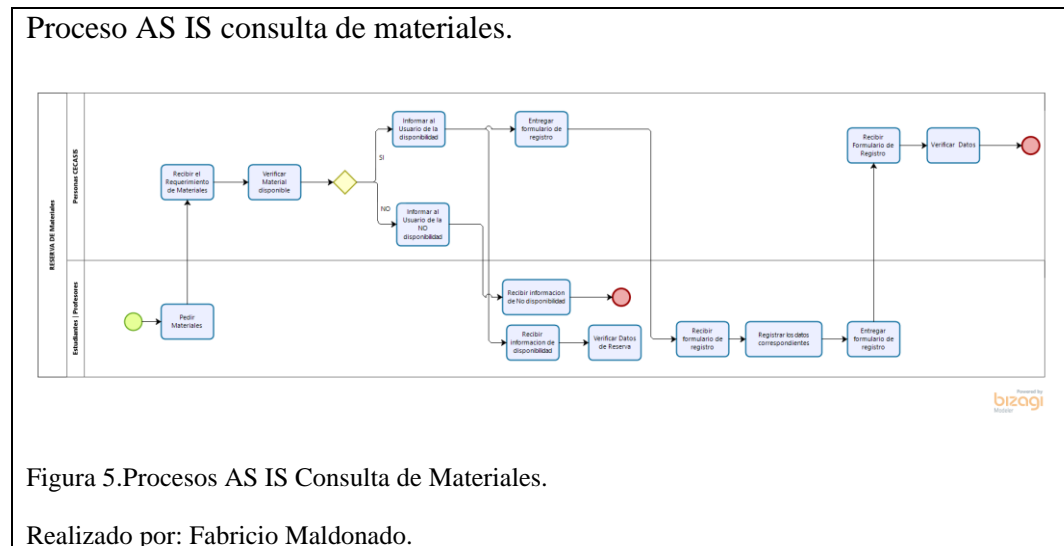
Figura 4. Ilustración. Procesos AS IS Reserva de laboratorios.

Realizado por. Fabricio Maldonado.

En el proceso notamos que se registra manualmente la verificación de los laboratorios disponibles, la entrega del formulario y los registros de reservas de laboratorios. Estas acciones requieren automatización.

2.2.1.2. Consulta de materiales.

La consulta de materiales se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.



En el proceso notamos que el Estudiante o Docentes deben acercarse presencialmente al CECASIS para verificar la disponibilidad de los materiales de préstamos.

2.2.1.3. Informe de mantenimiento.

El informe de mantenimiento se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

Procesos AS IS Informe de mantenimiento.

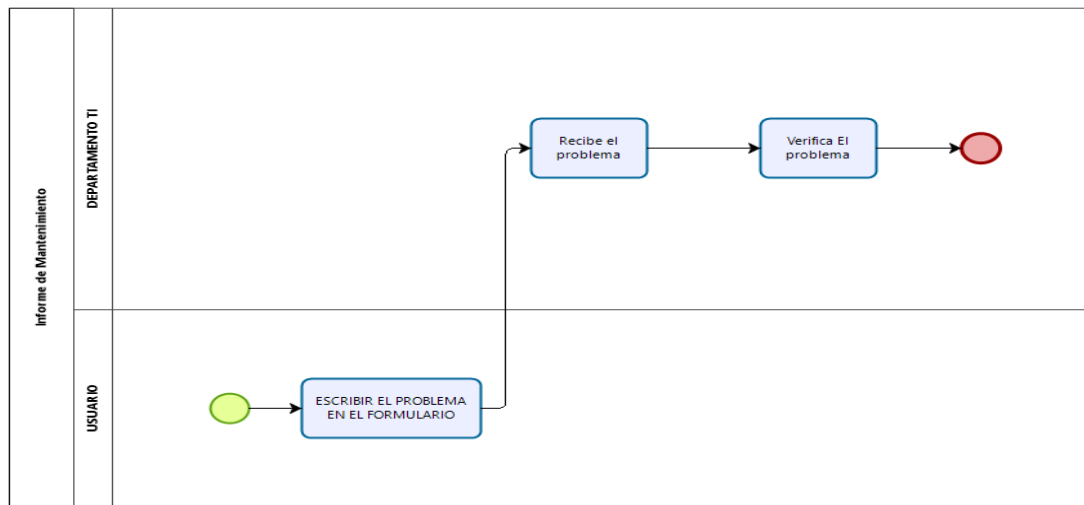


Figura 6. Procesos AS IS informe de Mantenimiento.

Realizado por: Fabricio Maldonado.

En el proceso notamos que se realiza un proceso manual por parte del Estudiante o Profesor al escribir el problema en el formulario. Existen tiempos muertos ya que una vez al día los técnicos revisan los problemas de mantenimientos.

2.2.2. Procesos to be.

Se utilizó los Procesos TO BE o “como debe ser” que es el proceso ya optimizado que se debe automatizar.

Se maneja los siguientes roles:

- Estudiante.
- Docente.
- Técnico.

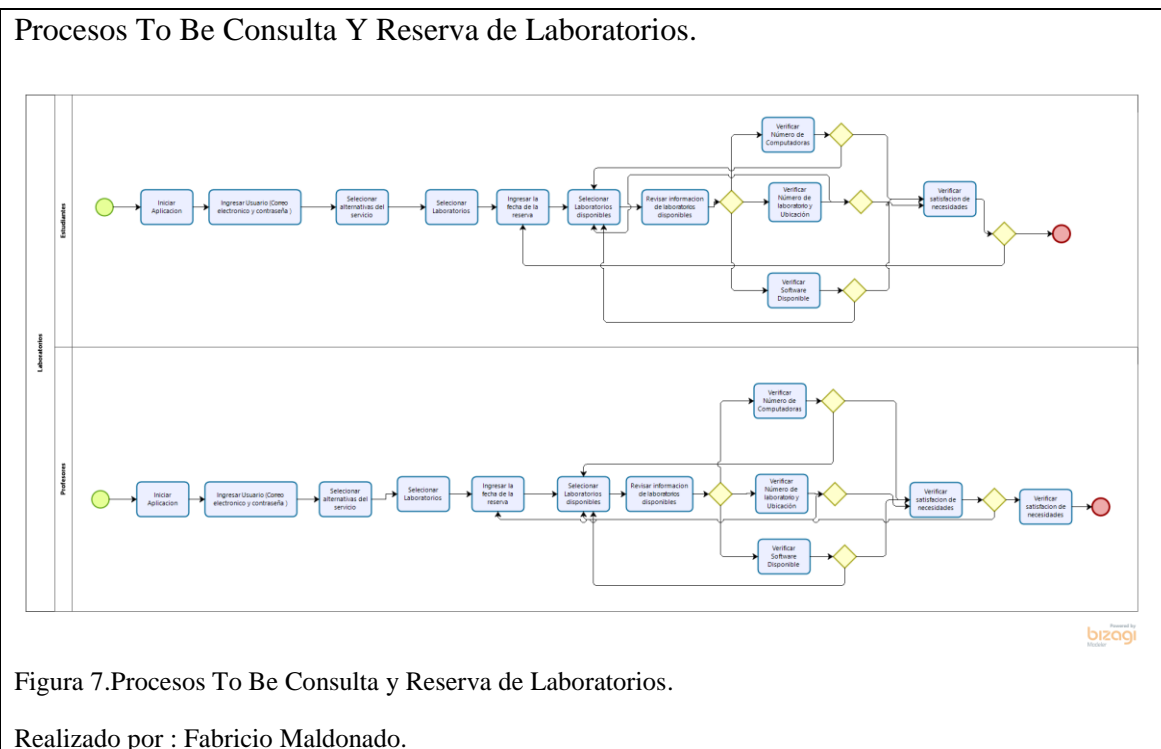
- Secretaria.

2.2.2.1. Consulta y reserva de laboratorios.

En la optimización de proceso de consulta y reserva de laboratorios se separó dos importantes tipos de usuarios:

Estudiantes: los estudiantes pueden ver los laboratorios disponibles y reservar los laboratorios con estado pendiente.

Docentes: los docentes pueden ver la reserva del estudiante y tiene la potestad de aceptar o rechazar la reserva, así mismo puede hacer sus propias reservaciones.

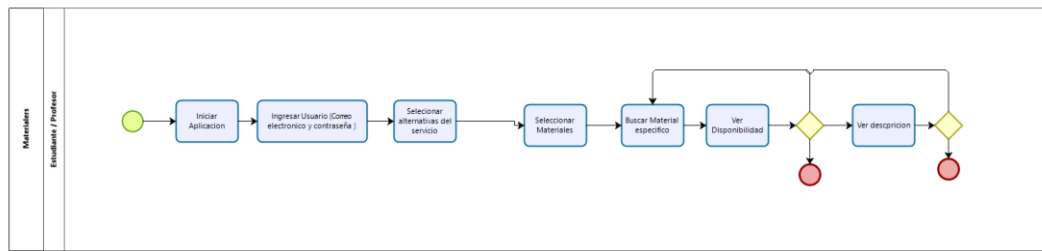


Tanto como docentes y estudiantes pueden hacer reservas para laboratorios, la diferencia es que para los estudiantes el estado se guarda como pendiente para que el docente sea quien apruebe la reserva.

2.2.2.2. Consulta de materiales.

La consulta de materiales se levantó junto con el dueño del proceso, en este caso con el personal del CECASIS. Se utilizó la notación BPMN para modelar dicho proceso.

Procesos de Consulta de Materiales.



Powered by
bizagi
Modeler

Figura

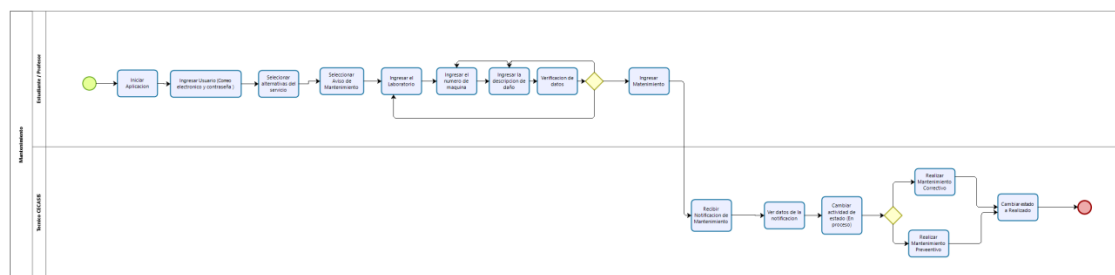
8. Proceso TO BE de Consulta de Materiales.

Realizado por Fabricio Maldonado.

2.2.2.3. Gestión de mantenimiento.

En la gestión del mantenimiento se eliminaron los procesos manuales los cuales fueron optimizados mediante registros de software.

Procesos TO BE de Gestion de Mantenimiento.



Powered by
bizagi
Modeler

Figura 9. Procesos TO BE de Gestion de Mantenimiento.

Realizado por: Fabricio Maldonado.

El estudiante o docente es encargado de registrar el incidente de los laboratorios.

El técnico asignado en ese horario será el encargado de solucionar el caso y cambiar de estado a la actividad para tener informado al personal del CECASIS sobre el incidente.

El técnico asignado será el encargado de solucionar y registrar la solución del trabajo realizado.

2.3. Requerimientos no funcionales.

La aplicación funciona y trabaja en diferentes ambientes (web y aplicación móvil) y hace uso de las siguientes aplicaciones:

Windows 10. Sistema Operativo en el que se instala los programas de desarrollo.

Apache, Version 2.4.33 Utilizado para alojar los servicios web de la cual consume la aplicación móvil.

PHP. Versión 5.6.36. Lenguaje de programación para levantar los servicios web y consumir de la base de datos de la aplicación web.

PostgreSQL. Versión 10. Utilizado para almacenar la información que será consumida por el servicio web y presentarla en la aplicación móvil.

Java 8. Leguaje de programación en la que se encuentra escrita la aplicación móvil , la cual debe ser compilada y ejecutada.

Android Studio 3.1.2. IDE de desarrollo en la cual mediante un Material Designer y programación en JAVA se genera y ejecuta la aplicación web.

JVM OpenJDK. Máquina Virtual en la que corre la aplicación en el dispositivo móvil.

2.4. Requerimientos funcionales.

2.4.1. Diagramas de casos de uso.

2.4.1.1. Caso de uso de autenticación e ingreso.

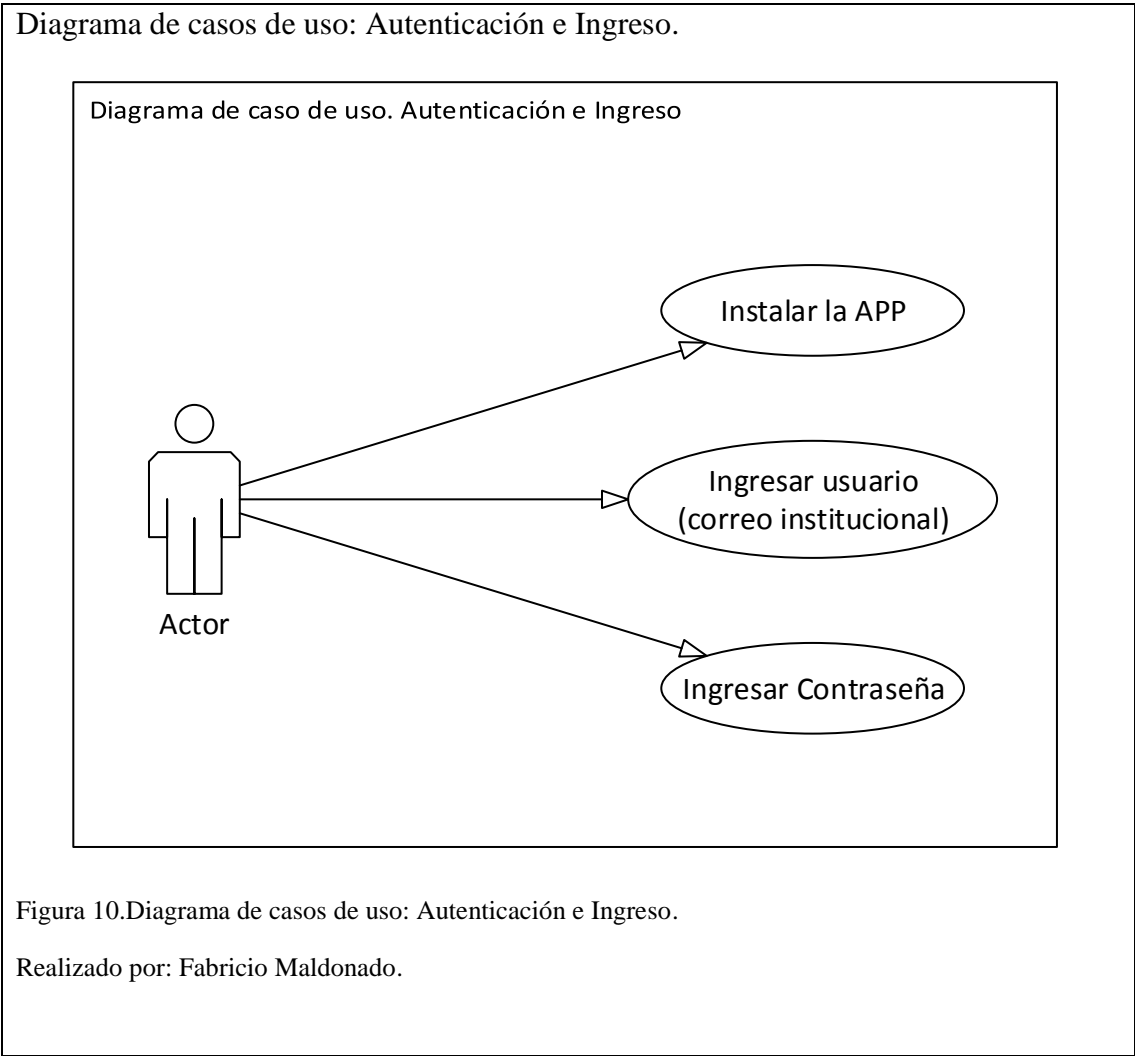
Caso de Uso Autenticación e Ingreso.

Caso de Uso #1.	Autenticación e ingreso.
Actores.	Usuario.
Camino Principal.	<ol style="list-style-type: none">1. Instalar la aplicación en el dispositivo móvil.2. Ingresar correo institucional.3. Ingresar contraseña institucional.
Precondiciones.	Para ingresar a la aplicación el servicio web debe regresar valores y no valores nulls.
Post-condiciones.	Instalación de la aplicación. El servidor de servicios debe ser inicializado.

Tabla 1. Caso de Uso Autenticación e Ingreso.

Realizado por: Fabricio Maldonado.

La autenticación y el ingreso es realizado consumiendo un servicio REST el cual está levantando en el servidor de aplicaciones que solicita dos variables, el correo institucional y la contraseña de usuario.



2.4.1.2. Diagrama de caso de uso de reservar laboratorio (docentes, estudiantes).

Caso de uso Reserva de Laboratorios.	
Caso de Uso #2.	Autenticación e ingreso.
Actores.	Docente / Estudiantes.

Camino Principal.	<ol style="list-style-type: none"> 1. Ingresar a opciones en el panel izquierdo superior de la pantalla. 2. Seleccionar la opción Laboratorios. 3. Seleccionar la fecha de reserva. 4. Seleccionar la hora de reserva. 5. Seleccionar el número de laboratorios. 6. Seleccionar el software requerido. 7. Buscar laboratorios disponibles. 8. Ver y revisar el laboratorio disponible. 9. Reservar el laboratorio disponible.
Precondiciones.	Acceder a la cuenta de usuario y autenticarse como docente de la Universidad Politécnica Salesiana.
Post-condiciones.	<p>Instalación de la aplicación.</p> <p>El servidor de servicios debe ser inicializado.</p> <p>Disponibilidad de laboratorios.</p>

Tabla 2. Caso de uso Reserva de Laboratorios.

Realizado por Fabricio Maldonado.

Diagrama de caso de uso. Reserva de Laboratorios

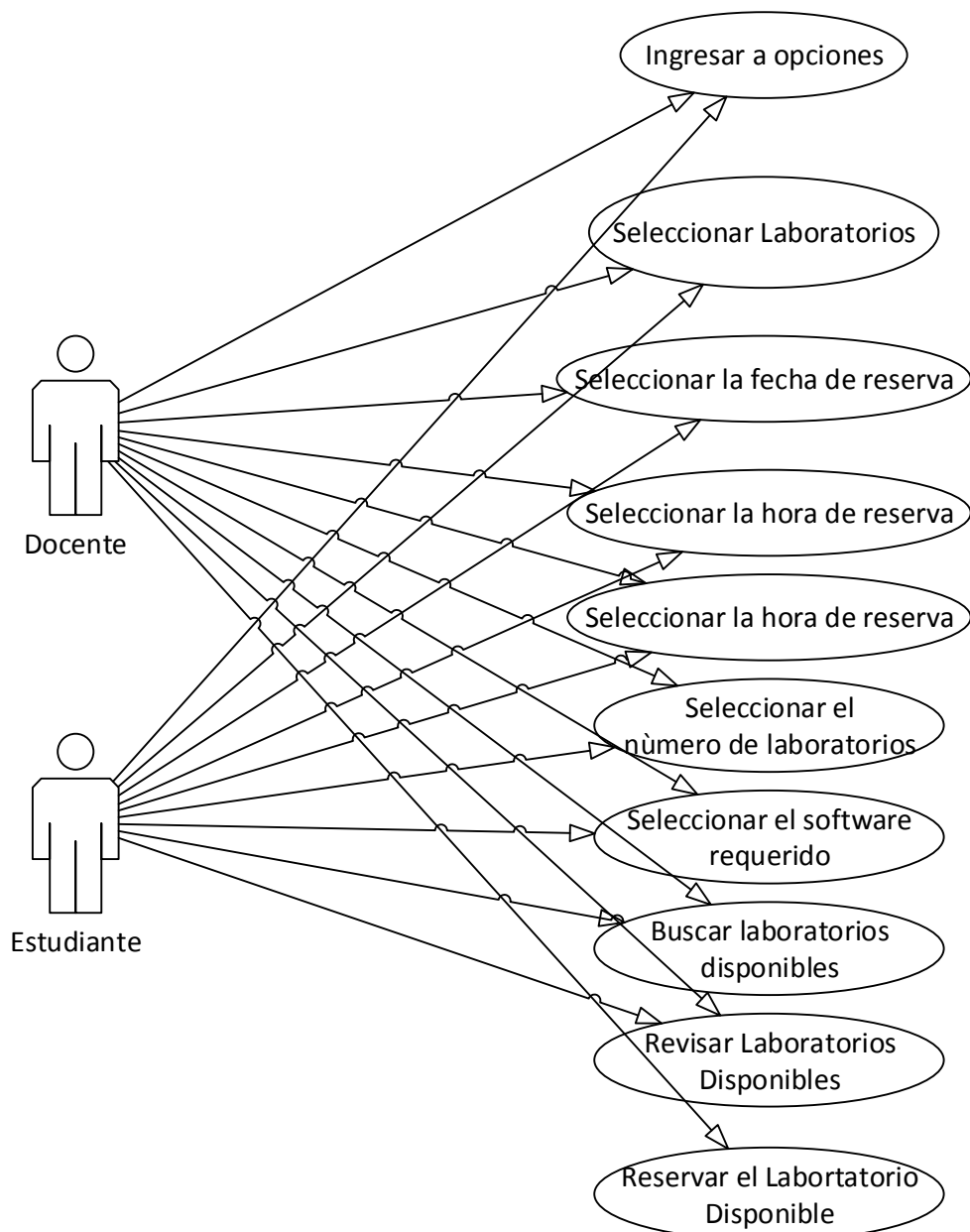


Figura 11. Diagrama de casos de Uso: Reserva de Laboratorio.

Realizado por: Fabricio Maldonado.

La diferencia entre los dos actores es que los estudiantes tienen la potestad de reservar un laboratorio en estado “Pendiente” mientras el tutor responsable asignado no lo apruebe. El docente tiene la potestad de reservar un laboratorio en estado “Reservado”

2.4.1.3. Diagrama de caso de uso de consulta de materiales.

Diagrama de caso de uso: consulta de materiales.	
Caso de Uso #3.	Autenticación e ingreso.
Actores.	Usuarios.
Camino Principal.	<ol style="list-style-type: none"> 1. Ingresar a Opciones. 2. Ingresar a Materiales. 3. Dar click en el botón buscar materiales. 4. Escribir el material requerido. 5. Click en el material requerido. 6. Visualizar el material requerido. 7. Verificar disponibilidad.
Precondiciones.	Acceder a la cuenta de usuario y autenticarse como docente o estudiante de la Universidad Politécnica Salesiana.
Post-condiciones.	Instalación de la aplicación. El servidor de servicios debe ser inicializado.

Tabla 3. Diagrama de caso de uso: consulta de materiales.

Realizado por: Fabricio Maldonado.

Diagrama de casos de uso: Consulta de Materiales.

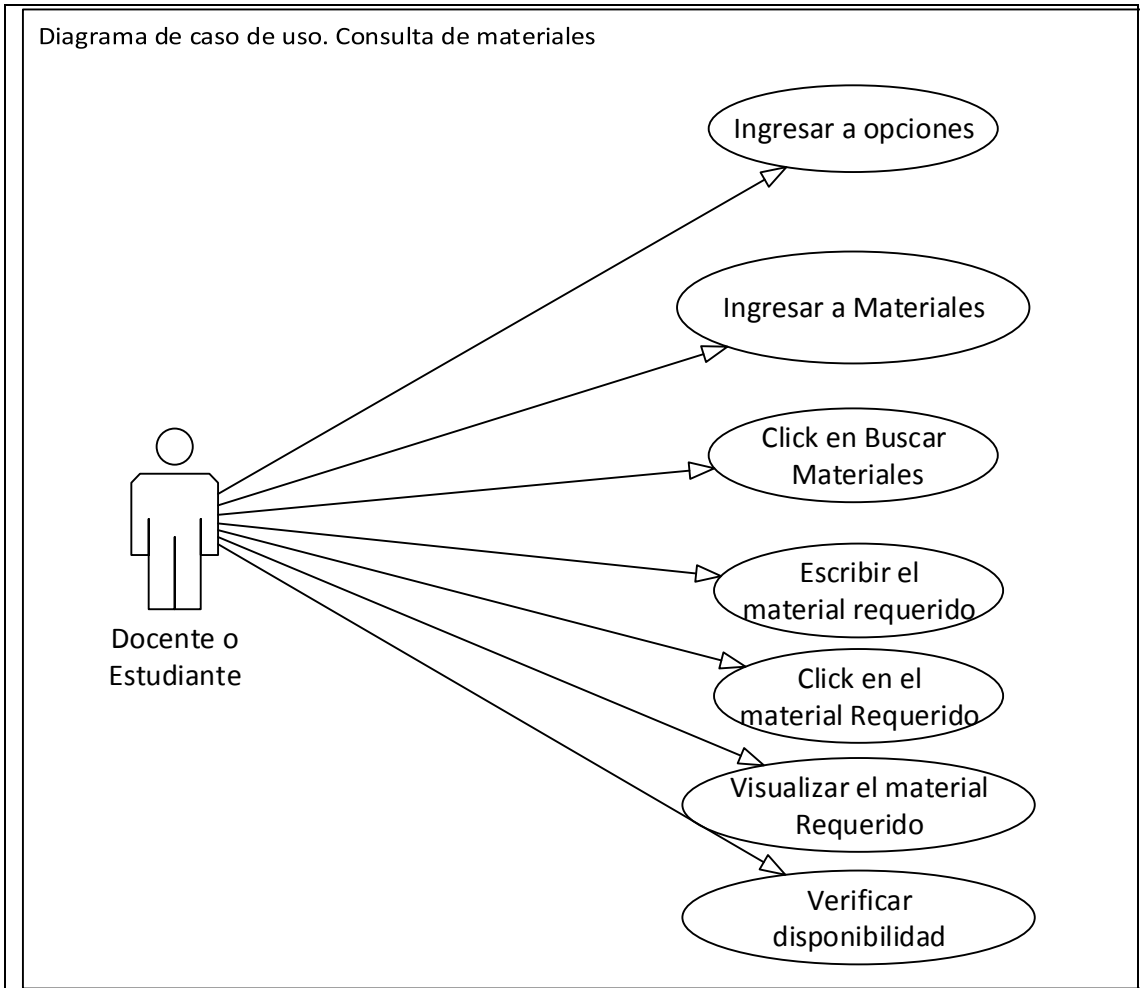


Figura 12. Diagrama de casos de uso: Consulta de Materiales.

Realizado por: Fabricio Maldonado.

2.4.1.4. Diagrama de casos de uso de gestión de mantenimiento.

Diagrama de Casos de Uso: Gestión de Mantenimiento.		
Caso de Uso #4.	Autenticación e ingreso.	
Actores.	Docente / Estudiantes.	Técnico/ Secretaria.
Camino Principal.	1. Ingresar a Opciones. 2. Ingresar a Mantenimiento.	1. Ingresar a Opciones. 2. Ingresar a Mantenimiento.

	<p>3. Ingresar a Incidentes.</p> <p>4. Ingresar tipo de incidentes.</p> <p>5. Ingresar descripción de incidentes.</p> <p>6. Tomar foto del incidente.</p> <p>7. Guardar incidente.</p> <p>8. Ver el estado incidente guardado.</p> <p>9. Ingresar a Registro de Incidentes.</p> <p>10. Revisar estado del incidente.</p>	<p>3. Ingresar a Incidentes.</p> <p>4. Ingresar tipo de incidentes.</p> <p>5. Ingresar descripción de incidentes.</p> <p>6. Tomar foto del incidente.</p> <p>7. Guardar incidente.</p> <p>8. Ver el estado incidente guardado.</p> <p>9. Ingresar a Registro de Incidentes.</p> <p>10. Cambiar de estado al incidente.</p> <p>Revisar estado del incidente.</p>
Precondiciones.	Acceder a la cuenta de usuario y autenticarse como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana.	Acceder a la cuenta de usuario y autenticarse como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana.
Post-condiciones.	Instalación de la aplicación.	Instalación de la aplicación.

	El servidor de servicios debe ser inicializado.	El servidor de servicios debe ser inicializado.
--	---	---

Tabla 4. Diagrama de Casos de Uso: Gestión de Mantenimiento.

Realizado por: Fabricio Maldonado.

La Gestión del incidente consume Servicios REST los cuales deben estar levantados en el servidor de aplicaciones.

Los servicios que se deben consumir son: ingreso de incidentes, consulta de incidentes, modificación de estado de incidentes.

Diagrama de Casos de Uso: Consulta de Materiales.

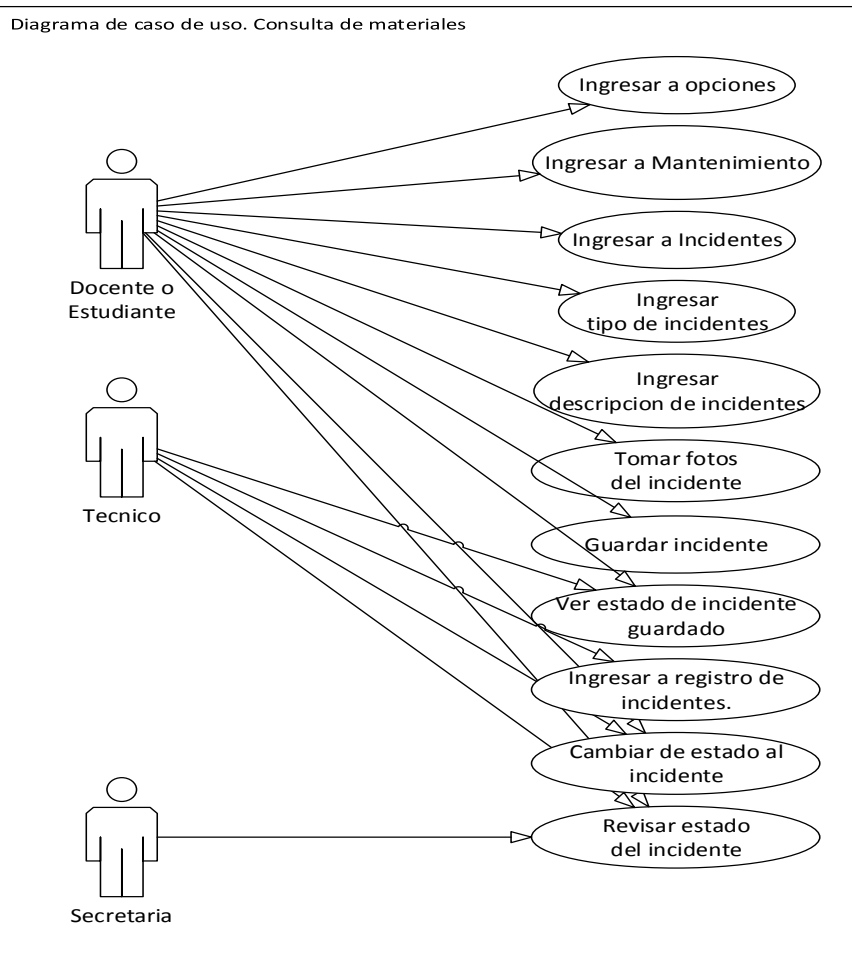


Figura 13. Diagrama de Casos de Uso: Consulta de Materiales.

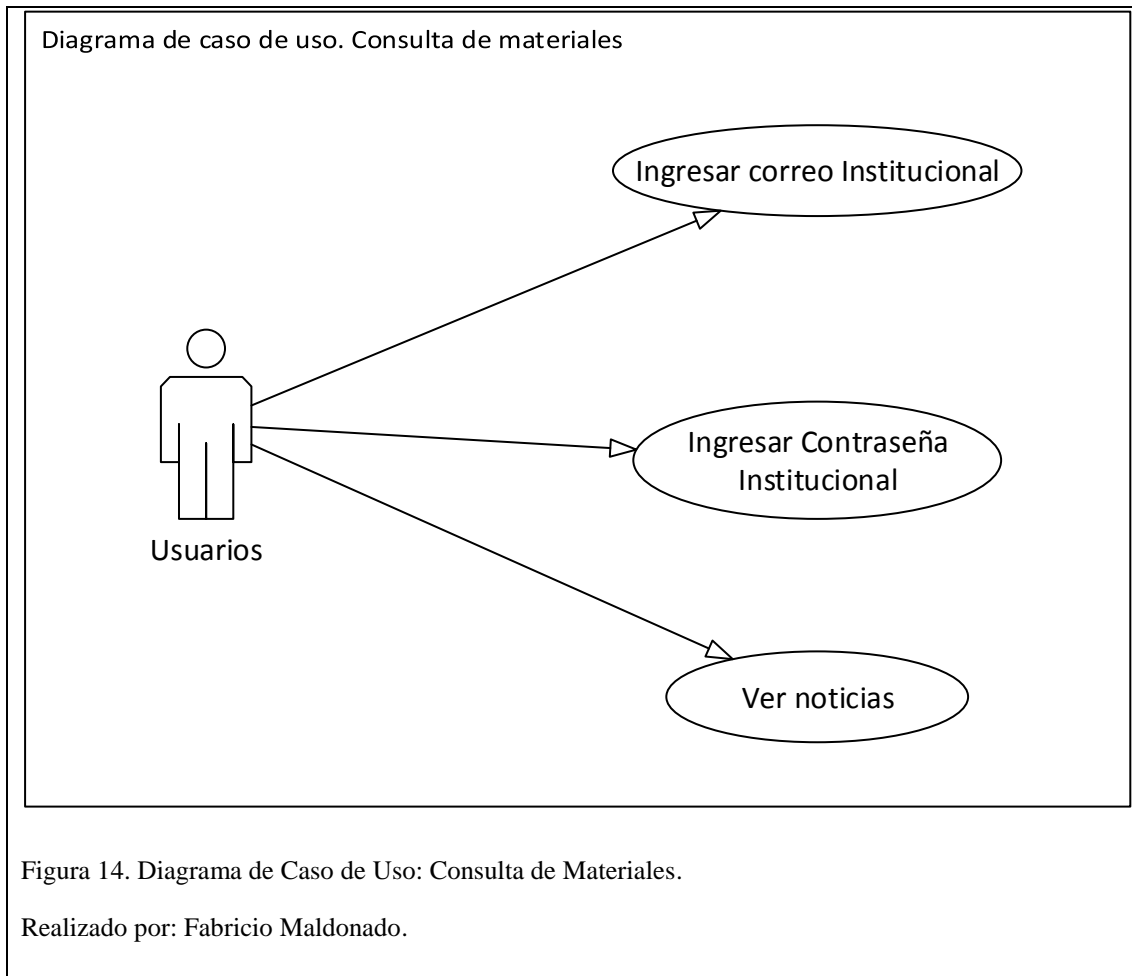
Realizado por: Fabricio Maldonado.

El usuario Secretaria tiene acceso para supervisar el estado del incidente y que ayude a la toma de decisiones.

2.4.1.5. Diagrama de caso de uso de noticias.

Diagrama de caos de Uso: Noticias.	
Caso de Uso #4.	Autenticación e ingreso.
Actores.	Usuarios.
Camino Principal.	<ol style="list-style-type: none"> 1. Ingresar correo institucional. 2. Ingresar contraseña institucional. 3. Ver las noticias.
Precondiciones.	Acceder a la cuenta de usuario y autenticarse como docente, estudiante, técnico o secretaria de la Universidad Politécnica Salesiana.
Post-condiciones.	<p>Instalación de la aplicación.</p> <p>El servidor de servicios debe estar inicializado.</p>
Tabla 5. Diagrama de caos de Uso: Noticias.	
Realizado por: Fabricio Maldonado.	

Diagrama de Caso de Uso: Consulta de Materiales.
--



La consulta de materiales se lo realiza como método informativo para evitar la consulta presencial sobre la disponibilidad de dichos materiales.

Base de Datos del Sistema Web.

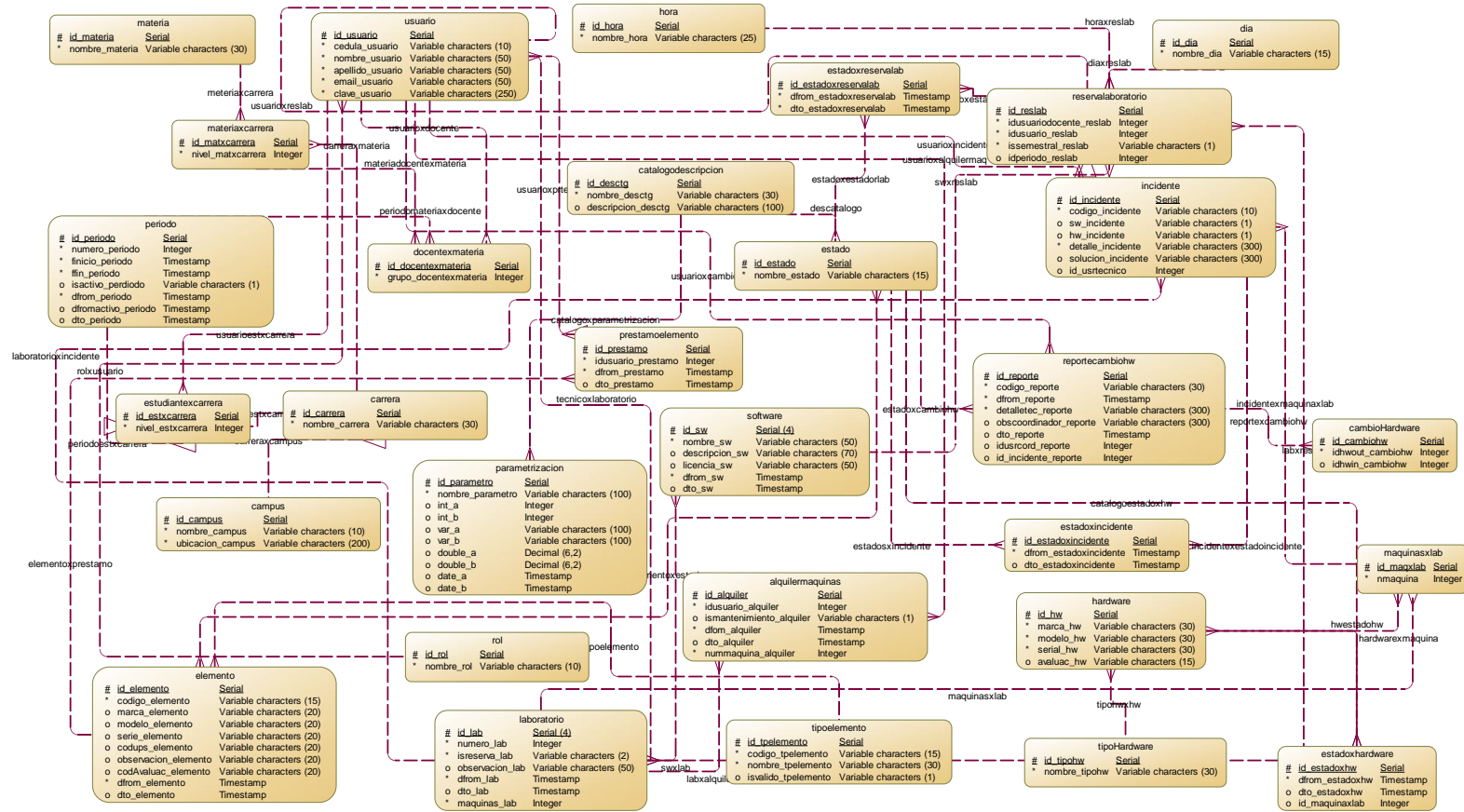


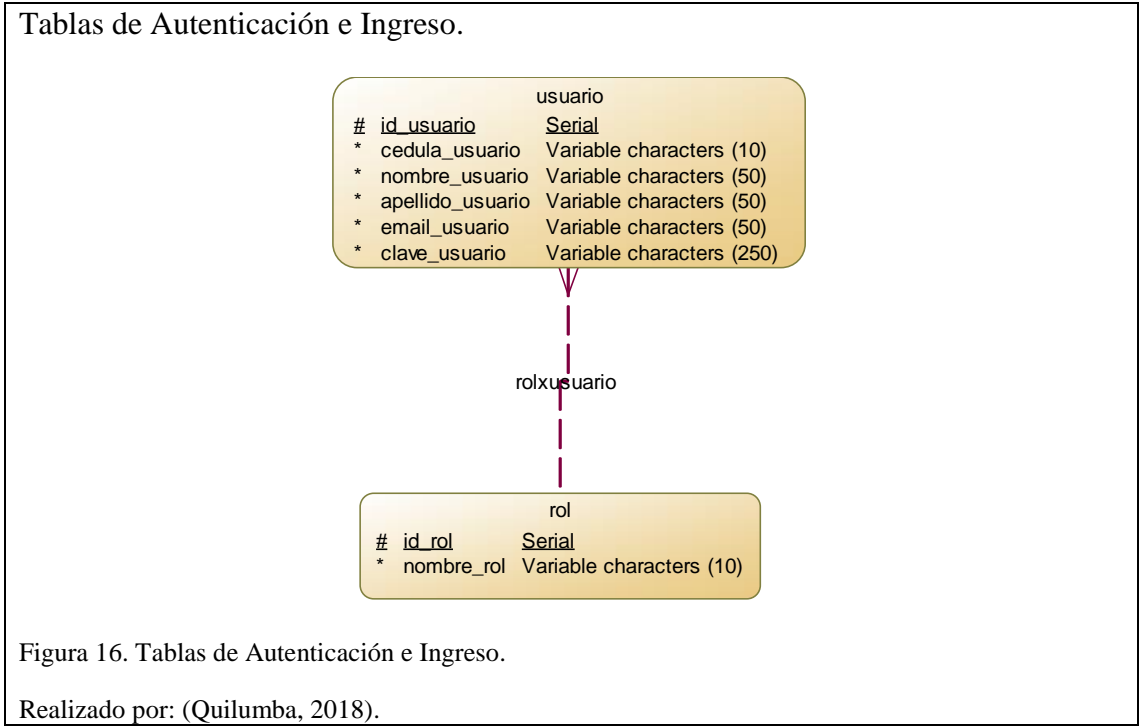
Figura 15. Base de Datos del Sistema Web.

Realizado por: (Quilumba, 2018).

La base de datos se encuentra actualmente en funcionamiento con el Sistema Web del Cecasis el cual gestiona Laboratorios, Materiales, Incidentes, Reservas, etc. El cual se está desarrollando junto a este proyecto.

Se está consumiendo servicios Web que tienen comunicación con las tablas necesarias para el funcionamiento de la aplicación y lograr su integración sin necesidad de conectarse a la base directamente.

2.5.1. Diseño de base de datos de autenticación e ingreso.



Las tablas esenciales para realizar la autenticación son rol y usuarios, ya que la aplicación móvil maneja sesiones y habilitar opciones de acuerdo al rol del usuario.

2.5.2. Diseño de base de datos de reserva de laboratorio.

Tablas de Reserva de Laboratorios.

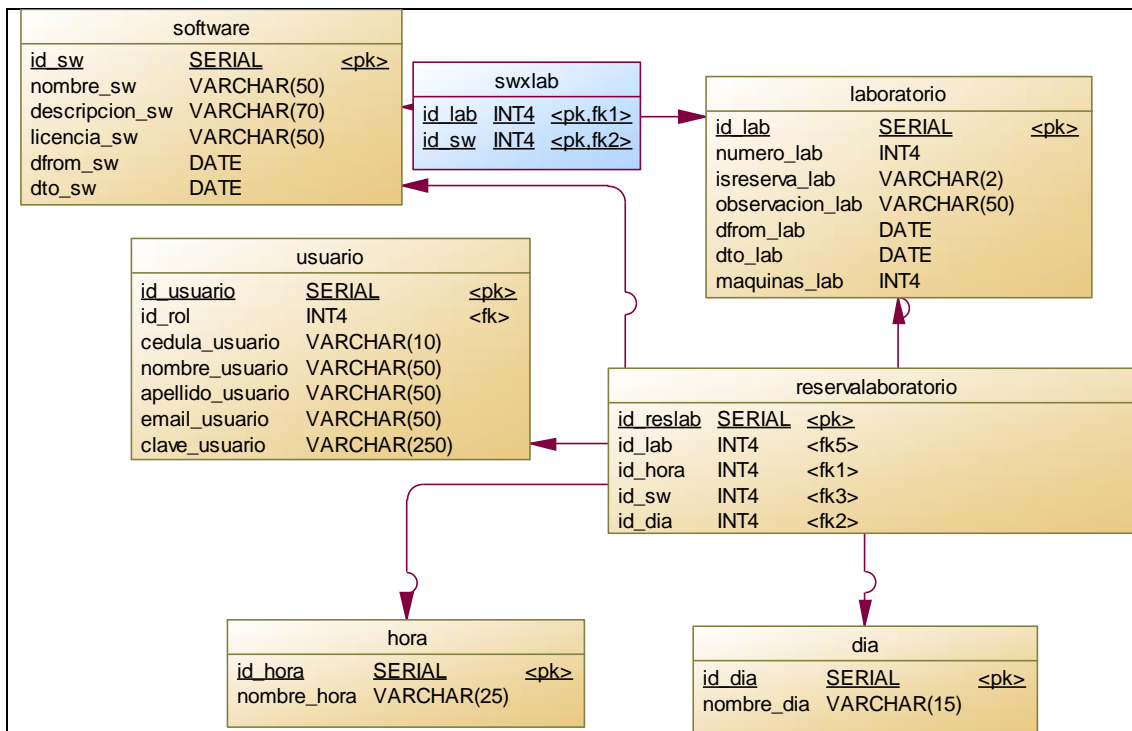


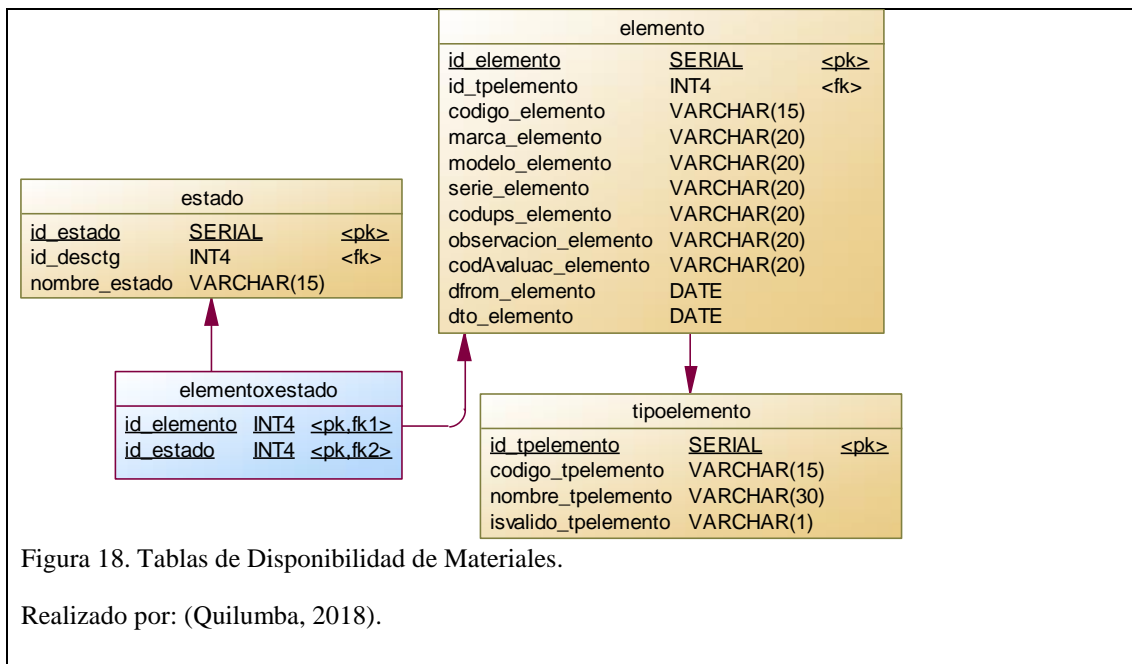
Figura 17. Tablas de Reserva de Laboratorios.

Realizado por: (Quilumba, 2018).

Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Laboratorios la cual realiza el filtrado de laboratorio por fecha, número de máquinas y el software que se encuentra instalado en los laboratorios.

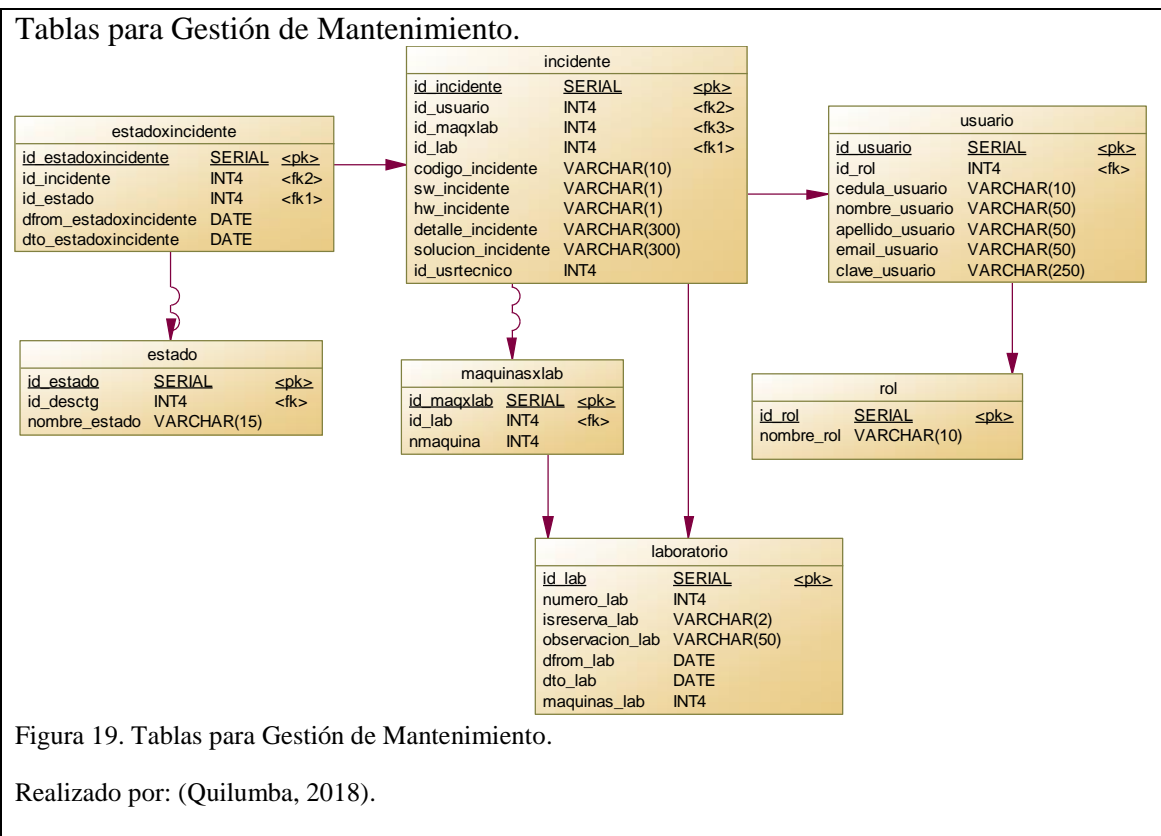
2.5.3. Diseño de base de datos de disponibilidad de materiales.

Tablas de Disponibilidad de Materiales.



Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Materiales la cual realiza la búsqueda y verificación de disponibilidad y estado de los materiales.

2.5.4. Diseño de base de datos de gestión de mantenimiento (incidentes).



Se muestran en la imagen las tablas esenciales para el funcionamiento de la subsección Mantenimiento, la cual se encarga de registrar un incidente o daño de la maquina o máquinas de los laboratorios del CECASIS.

2.6. Diseño de autenticación de aplicación mediante tokens.

La implementación de la seguridad entra aplicaciones se lo realizo mediante una autenticación del dispositivo previa en el sistema web. Por lo cual se utilizó una identificación del dispositivo encriptada para poder hacer el registro.

Sistema de autenticación entra aplicaciones.

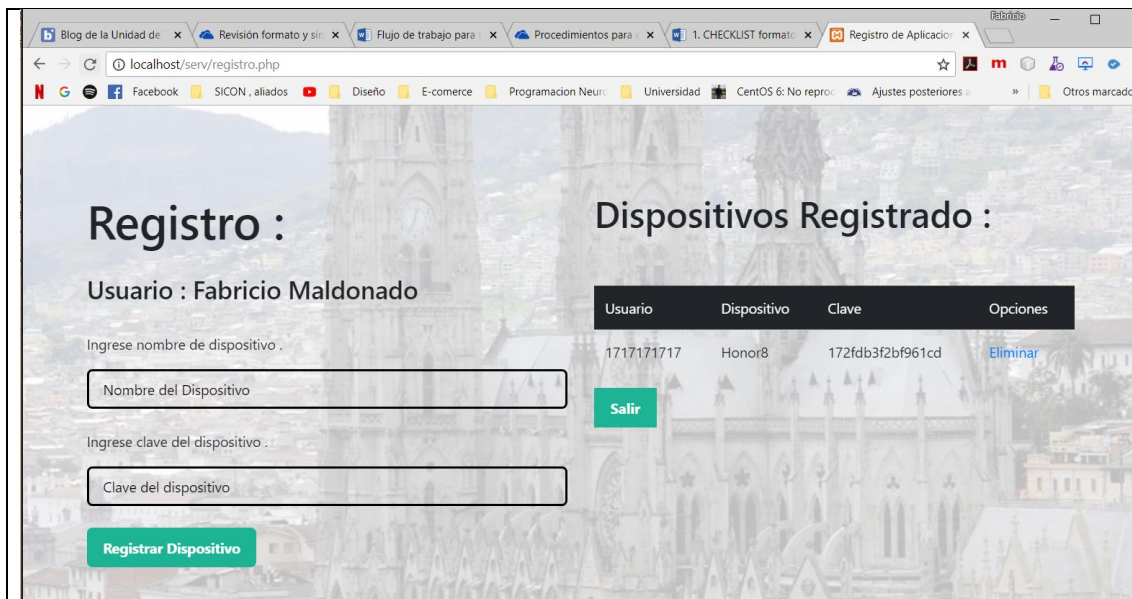


Figura 20. Sistema de autenticación entra aplicaciones.

Realizado por Fabricio Maldonado.

Una vez realizado el registro el servidor de aplicaciones reconoce al dispositivo de confianza y el dispositivo móvil ya puede iniciar sesión.

2.6.1. Diseño del servicio web.

Los servicios web tienen que estar constantemente levantados para ser consumidos, así mismo para lograr la integración con la aplicación web la cual va a gestión de manera más precisa Laboratorios, Materiales e Incidentes.

2.6.1.1. Consumo de servicios mediante tokens.

Para ser consumido un servicio web se necesita un token previamente generado al momento de lograr un inicio de sesión correcto.

El servidor de aplicaciones es el encargado de generar el token mediante un algoritmo de encriptación previamente establecido.

Código para generar un token único.

```
function getToken()
{
$hostname = explode(".", gethostbyaddr($_SERVER['REMOTE_ADDR']));
$nommaque=$hostname[0];
$token = hash('sha512',$nommaque);

return $token;
}
?>
```

Figura 21. Código para generar un token único.

Realizado por Fabricio Maldonado.

Mientras el token no se genere y sea asignado a un dispositivo móvil, no se podrá consumir ningún servicio web, para así evitar cualquier tipo de ingreso no permitido.

Código de comparación entre el token asignado y el token del servidor de aplicaciones.

```
<?php
session_start();

include("conexion.php");
$db = getDB();
$token = getToken();
if(isset($_REQUEST['key'])){
    $tokenapp = $_REQUEST['key'];

    if($token == $tokenapp){

        $res=$db->query("select * from laboratorio");
```

Figura 22. Código de comparación entre el token asignado y el token del servidor de aplicaciones.

Realizado por Fabricio Maldonado.

2.7. Diseño de clases.

A continuación, se presenta los diagramas de clases de la aplicación móvil.

2.7.1. Diagrama de clases de la aplicación móvil.

El Diagrama de clases nos facilita ver la estructura de la aplicación móvil, con sus atributos, métodos y operaciones.

2.7.1.1. Diagrama de clases de autenticación e ingreso y noticias.

Diagrama de Clases: Autenticación e Ingreso.

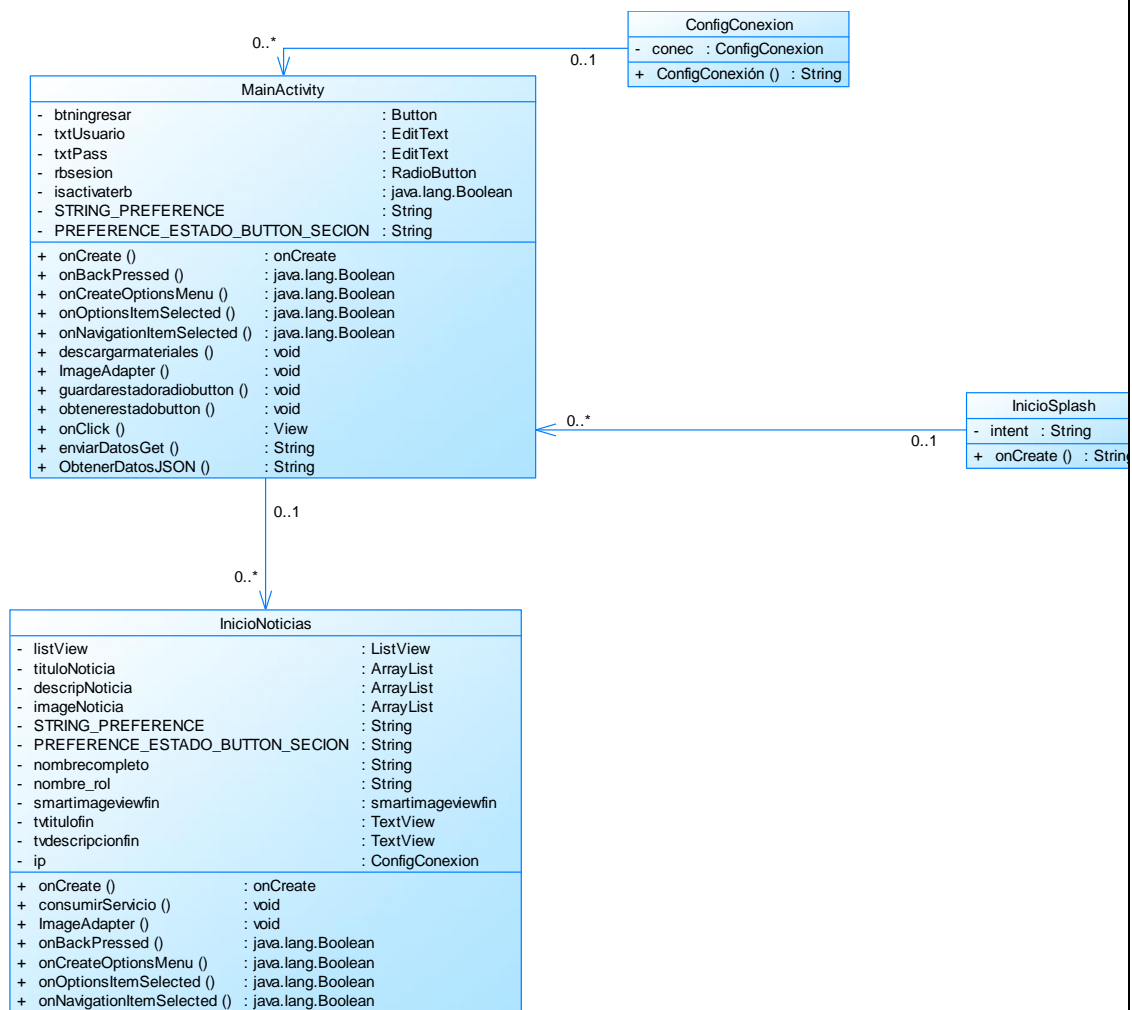


Figura 23. Diagrama de Clases Autenticación e Ingreso.

Realizado por: Fabricio Maldonado.

La aplicación móvil comienza cuanta con una animación inicial para proceder con el Login de la aplicación, la clase principal **MainActivity** llama a un objeto **ConfigConexion** para la autenticación del usuario.

La clase principal **MainActivity** autentifica el usuario por medio de servicios web e inicia la clase **InicioNoticias**, la cual, mediante servicios web, presenta la información de noticias recientes, esta clase carga los datos en forma asíncrona.

2.7.1.2. Diagrama de clases de reserva de laboratorios.

Diagrama de Clases de Reserva de Laboratorios.

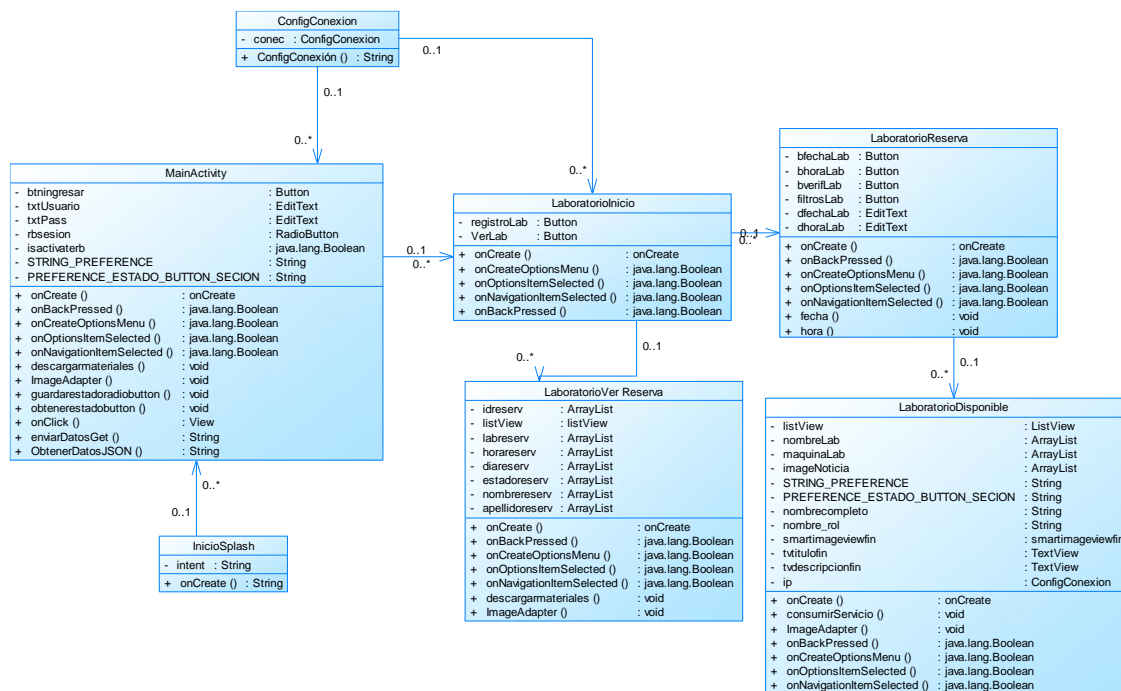


Figura 24. Diagrama de Clases de Reserva de laboratorios.

Realizado por: Fabricio Maldonado.

LaboratoriosInicio es la clase inicial de la subsección Laboratorios, la cual es llamada por medio de la clase principal MainActivity, esta clase llama a dos clases las cuales son:

- **LaboratoriosReserva:**
- Esta clase filtra y busca los laboratorios según la necesidad del usuario la cual da paso a **LaboratoriosDisponible** para realizar la confirmación de la reserva de laboratorios.

- **LaboratoriosVerReserva:**
- Esta clase visualiza los laboratorios reservado por el usuario.

2.7.1.3. Diagrama de clases de disponibilidad de materiales.

Diagrama de Clases de Disponibilidad de Materiales.

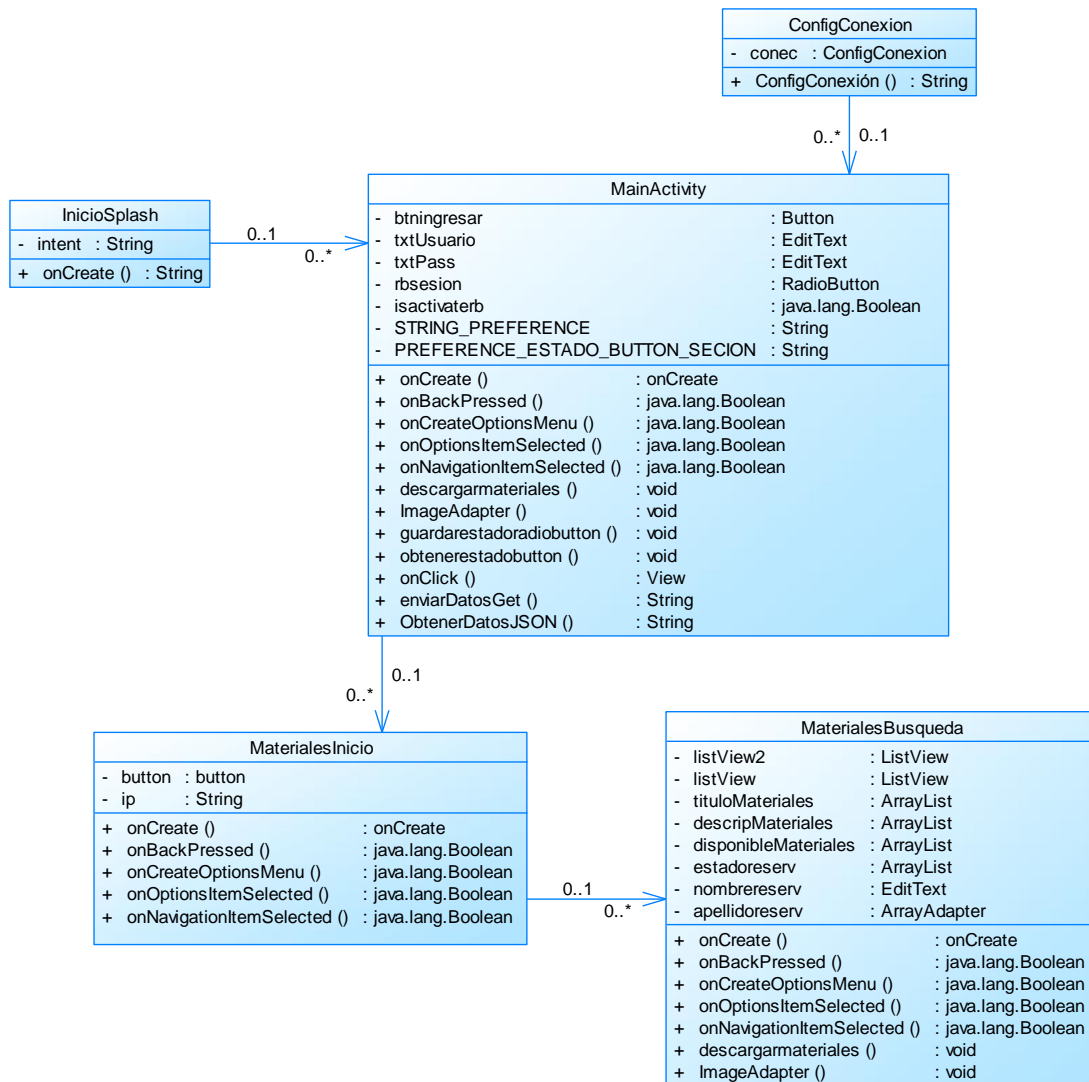


Figura 25. Diagrama de clases de Materiales.

Realizado por: Fabricio Maldonado.

MaterialesInicio es la clase inicial de la subsección Materiales la cual es llamada por medio de la clase principal **MainActivity**.

La clase **MaterialesInicio** llama **MaterialesBusqueda** para proceder con la búsqueda de materiales y así mismo la disponibilidad de cada uno de ellos.

2.7.1.4. Diagrama de clases de gestión de mantenimiento (incidentes).

Diagrama de Clases de Gestión de Mantenimiento.

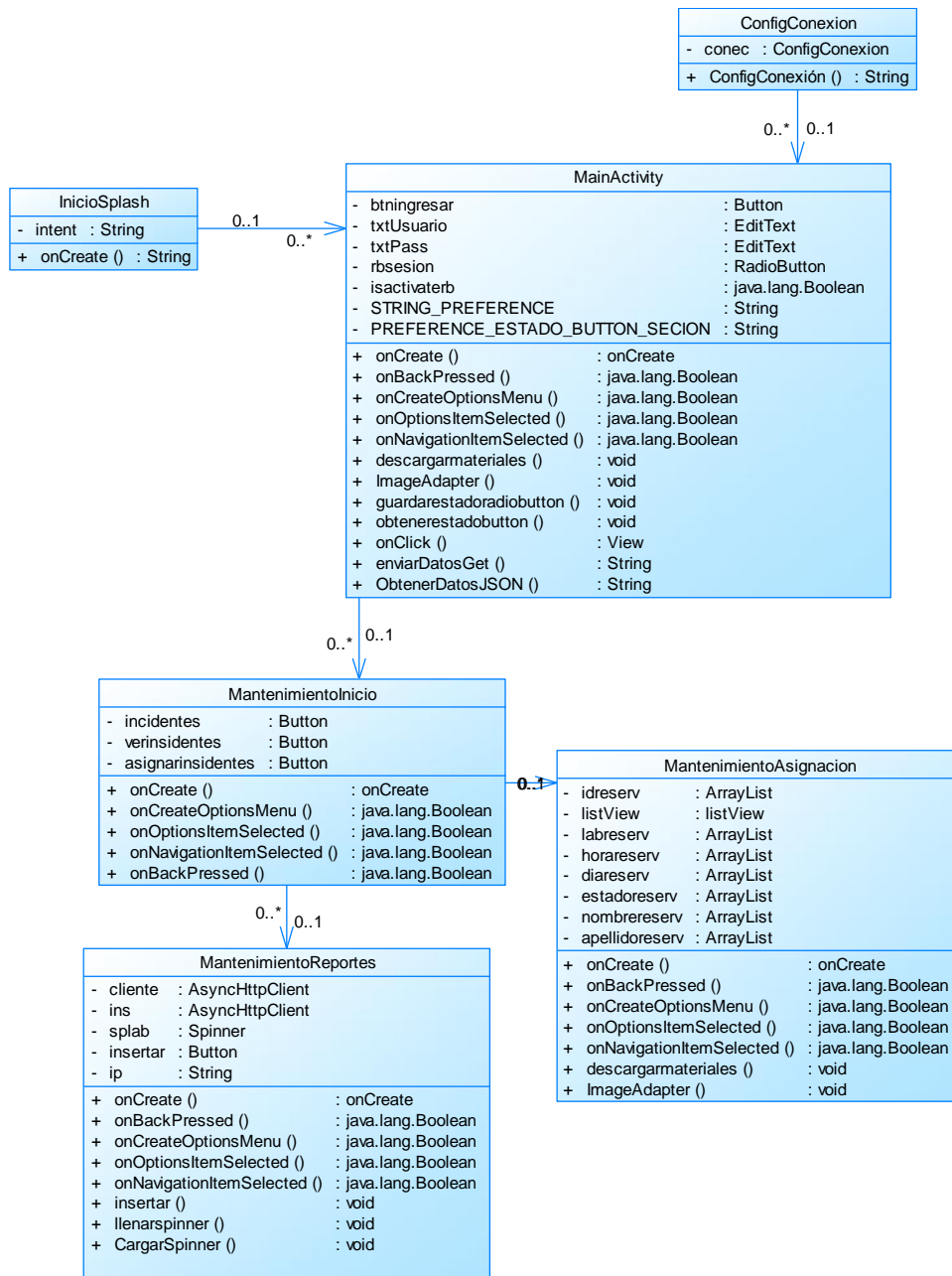


Figura 26. Diagrama de Clases de Gestión de Mantenimiento.

Realizado por: Fabricio Maldonado.

MantenimientoInicio es la clase inicial de la subsección Mantenimiento la cual es llamada por medio de la clase principal **MainActivity**.

MantenimientoInicio llama a dos clases:

MantenimientoAsignacion: en la cual se ingresa un incidente de laboratorios y se asigna automáticamente al técnico designado en el horario específico.

MantenimientoReportes: esta clase genera reportes y seguimiento a los incidentes para que el usuario pueda ver el estado del incidente ingresado.

2.8. Diagrama de despliegue.

El sistema web y el sistema para dispositivos móviles son integrados mediante servicios web.

Diagrama de despliegue de la aplicación web y móvil.

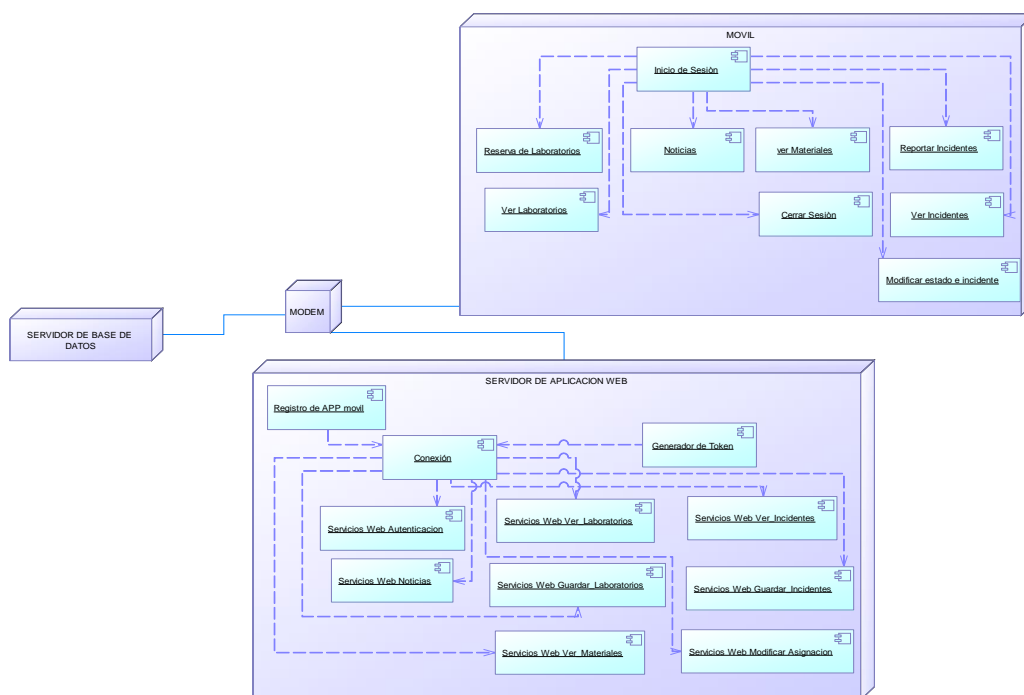


Figura 27. Diagrama de despliegue de la aplicación web y móvil.

Realizado por Fabricio Maldonado.

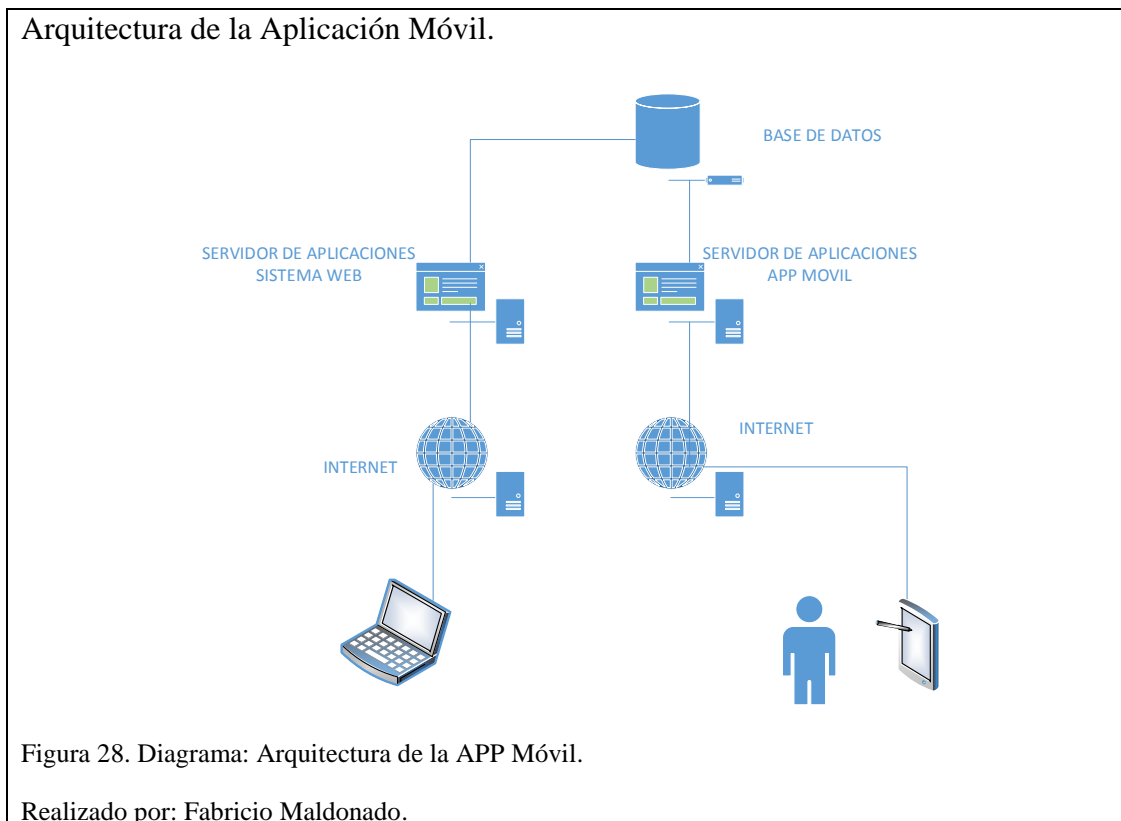
Capítulo 3

3. Construcción.

Para el desarrollo de la aplicación móvil se consideran las 5 etapas mencionadas en el capítulo de análisis y diseño: Autenticación e ingreso, Noticias Actuales, Consulta y Reserva de Laboratorios, Consulta de disponibilidad de materiales y Gestión de Incidentes.

3.1. Construcción de la aplicación móvil.

3.1.1. Arquitectura de la aplicación.



La aplicación móvil se conecta por medio de internet a un servidor de aplicaciones en el cual actualmente se levantan los servicios web necesarios para el funcionamiento de la misma.

3.2. Construcción de interfaz y funcionalidad.

El diseño de las interfaces de la aplicación está realizado mediante el estándar de Material Designer que propone una estructura de xml para los diferentes componentes de la aplicación.

3.2.1. Construcción de la autenticación e ingreso.

La interfaz se realizó con un modelamiento basado en xml el que dispone de los siguientes componentes:

Componentes utilizados en Autenticación e Ingreso.

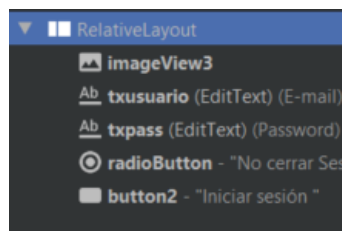


Figura 29. Componentes utilizados en Autenticación e Ingreso.

Realizado por: Fabricio Maldonado.

La característica más importante es la asignación del id. Este identificador es el encargado de darle una entidad al componente para luego ser llamado por el controlador de la aplicación.

Interfaz de Autenticación e Ingreso.

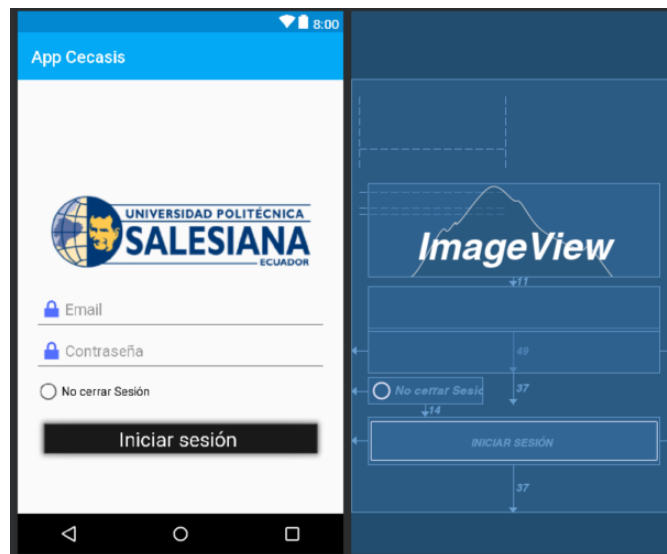


Figura 30. Interfaz de Autenticación e Ingreso.

Realizado por: Fabricio Maldonado.

El botón iniciar sesión crea un hilo el cual gestiona el envío de datos y la respuesta de datos por parte del servidor, así mismo guarda el estado del radio button para luego ser almacenado.

Código de Inicio de sesión.

```
public void onClick(View v) {  
    guardarestadoradiobutton();  
    Thread tr=new Thread(){  
        @Override  
        public void run() {
```

Figura 31. Código de Inicio de Sesión.

Realizado por: Fabricio Maldonado.

El radio button es almacenado en una variable de memoria interna, para poder identificar si el usuario ya tuvo un inicio de sesión y guardarlo.

Código Guardar estado de Inicio de Sesión.

```
public void guardarestadoradiobutton(){  
    SharedPreferences prefernece= getSharedPreferences(STRING_PREFERENCE.MODE_PRIVATE);  
    prefernece.edit().putBoolean(PREFERENCE_ESTADO_BUTTON_SECION,rbseccion.isChecked()).app
```

```
ly();  
}
```

Figura 32. Código Guardar estado de Inicio de Sesión.

Realizado por: Fabricio Maldonado.

Así mismo se guarda en la memoria interna el usuario y la contraseña de usuario que inició sesión.

Código Guardado en Variables de memoria interna.

```
SharedPreferences prefernece= getSharedPreferences(STRING_PREFERENCE,MODE_PRIVATE);  
prefernece.edit().putString("nombre",nombrecompleto).apply();  
prefernece.edit().putString("nombrerol",nombre_rol).apply();
```

Figura 33. Código Guardado en variables de memoria interna.

Realizado por: Fabricio Maldonado.

Se envía el usuario y contraseña al servicio web.

Código Guardado en Variables de memoria interna.

```
url = new URL("http://" + ip + "/serv/inicio.php?usu="+usu+"&pas="+pas+"");  
HttpURLConnection conection= (HttpURLConnection)url.openConnection();  
respuesta=conection.getResponseCode();
```

Figura 34. Código Guardado en variables de memoria interna.

Realizado por: Fabricio Maldonado.

Si el servicio web retorna una cadena JSON el usuario existe, en caso que el servicio web retorne un valor nulo el usuario no existe por lo cual no podrá iniciar sesión.

3.2.2. Construcción de noticias.

Para crear la sección de noticias se necesita 2 diferentes interfaces, una que consuma el servicio y una interfaz de estilo de la presentación de los datos.

Así mismo una interfaz que gestione la presentación del popup de noticias.

La estructura de la interfaz es:

Componentes utilizados en Noticias.

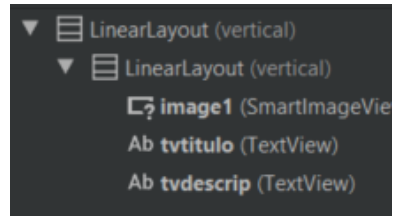


Figura 35. Componentes utilizados en Autenticación e Ingreso.

Realizado por: Fabricio Maldonado.

Interfaz de Noticias.

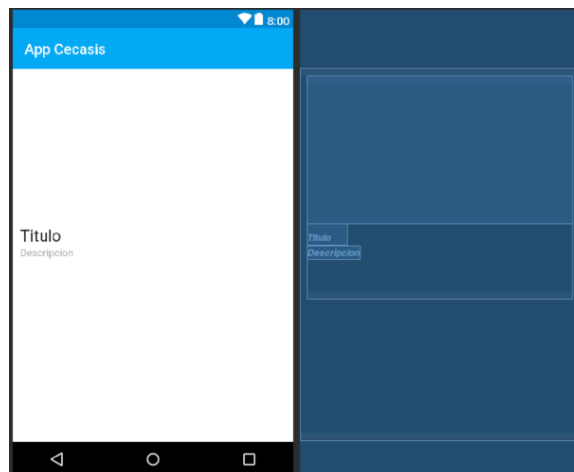


Figura 36. Interfaz de Noticias.

Realizado por: Fabricio Maldonado.

Se extrae los datos mediante una petición al servicio web, el cual retorna los datos de las noticias asincrónicamente, así mismo para una próxima carga, los datos se guardan en memoria caché.

Llamada Asincrónica a el servicio web noticias.

```
AsyncHttpClient client = new AsyncHttpClient();  
client.get("http://" + ip + "/serv/noticias.php", new AsyncHttpResponseHandler() {
```


Figura 37. Llamada Asincrónica a el servicio web noticias.

Realizado por: Fabricio Maldonado.

Para presentar la imagen en la pantalla se utiliza smartimageview la cual genera una imagen a partir de una ubicación y servicio web.

Para luego, tanto como los datos y las imágenes, sean estructuradas para su presentación en la interfaz.

Llamada Asincrónica a las imágenes en el servidor web.

```
String urlfinal="http://" + ip + "/serv/img/" + image.get(position).toString();  
Rect rect = new Rect (smartimageview.getLeft(),  
smartimageview.getTop(),smartimageview.getRight(),smartimageview.getBottom());  
smartimageview.setImageUrl(urlfinal,rect);
```

Figura 38. Llamada Asincrónica a las imágenes en el servidor web.

Realizado por. Fabricio Maldonado.

3.2.3. Construcción de módulo laboratorio.

El módulo de laboratorio está conformado por dos sub módulos los cuales fueron modelados con la siguiente estructura:

- Reserva de laboratorios.
- Ver laboratorios.

3.2.3.1. Construcción de reserva laboratorio.

Interfaz de Reserva de Laboratorios.

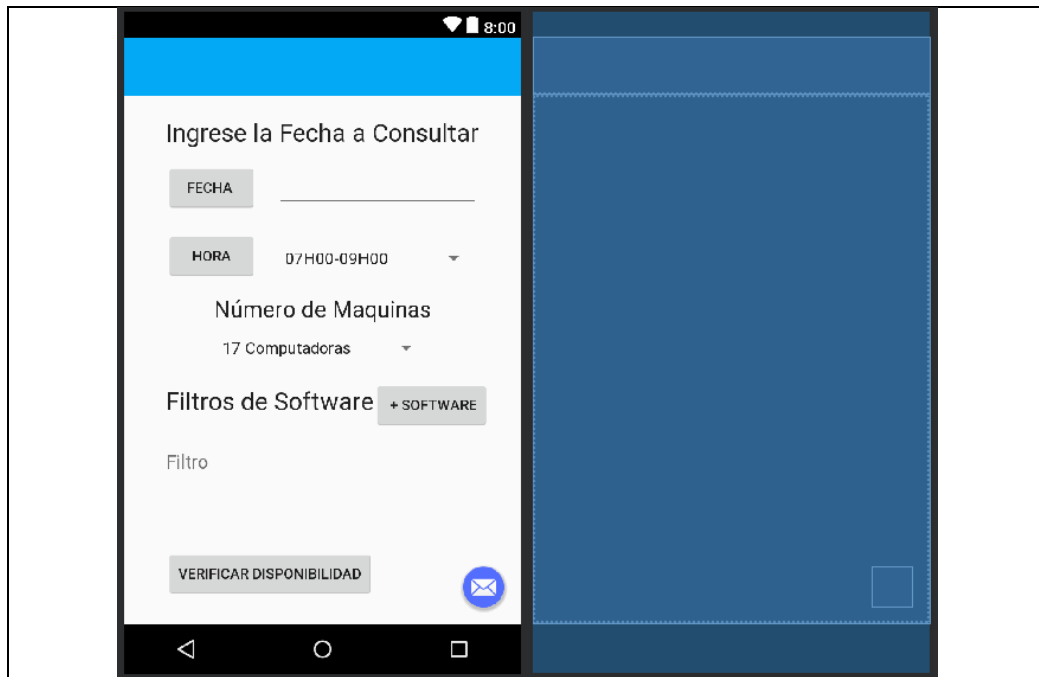


Figura 39. Interfaz de Reserva de Laboratorios.

Realizado por: Fabricio Maldonado.

Antes de hacer la reserva de laboratorios, se filtra la fecha, la hora, el número de máquinas y el software disponible por laboratorio. Para esto se utilizó calendar, una librería de google que hace uso del calendario del teléfono celular.

Llamada a DatePicker de Android.

```
DatePickerDialog datePickerDialog = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
@Override
public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
dfecha.setText(dayOfMonth+"/"+(monthOfYear+1)+"/"+year);
datePickerDialog.show();
}
```

Figura 40. Llamada a DatePicker de Android.

Realizado por: Fabricio Maldonado.

Se llama a un servicio web para ver las horas y el software disponible por laboratorio.

3.2.3.2. Construcción de ver laboratorios.

En este sub módulo se visualizan los laboratorios filtrados para continuar con la reserva según los requerimientos de usuario.

Interfaz de ver laboratorios.

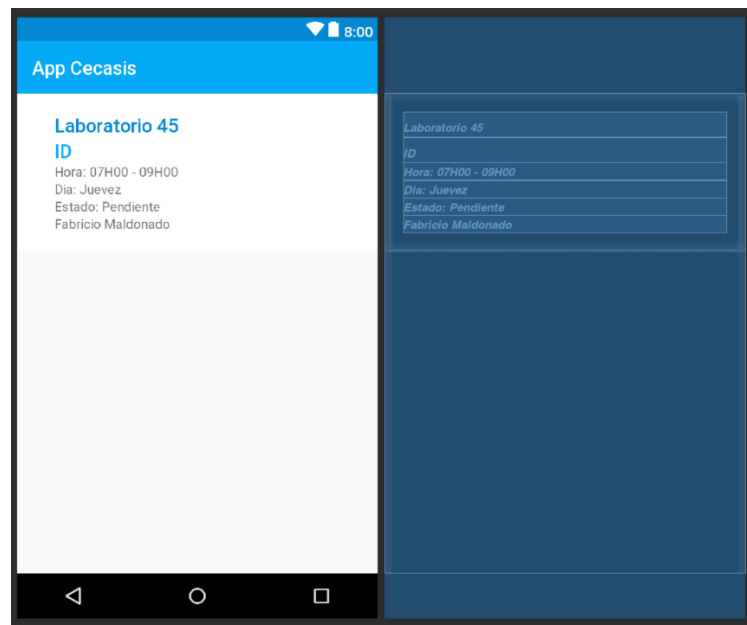


Figura 41.
de ver

Interfaz

laboratorios.

Realizado por: Fabricio Maldonado.

La estructura de la interfaz es:

Componentes utilizados en ver Laboratorios.

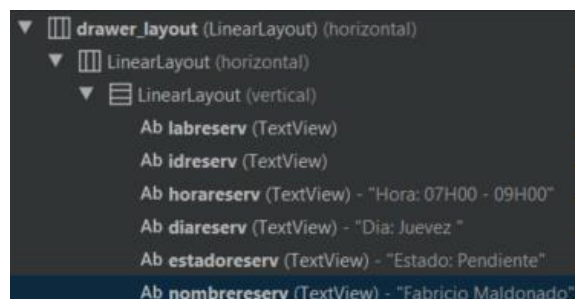


Figura 42. Componentes utilizados en ver Laboratorios.

Realizado por: Fabricio Maldonado.

Para visualizar los laboratorios reservados se debe implementar un ListView y sus respectivos ArrayList por cada dato visualizado en la interfaz:

Código Declaración de los Arrays para exponer en el ListView.

```
private ListView listView;  
  
ArrayList idreserv = new ArrayList();  
ArrayList labreserv = new ArrayList();  
ArrayList horaserv= new ArrayList();  
ArrayList diareserv= new ArrayList();  
ArrayList estadoreserv= new ArrayList();  
ArrayList nombreserv= new ArrayList();  
ArrayList apellidoserv= new ArrayList();
```

Figura 43. Código Declaración de los Arrays para exponer en el ListView.

Realizado por: Fabricio Maldonado.

3.2.4. Construcción de módulo materiales.

El modulo materiales, nos ayuda a buscar y verificar la disponibilidad de los materiales de préstamos en el CECASIS.

Interfaz de Consulta de Materiales.

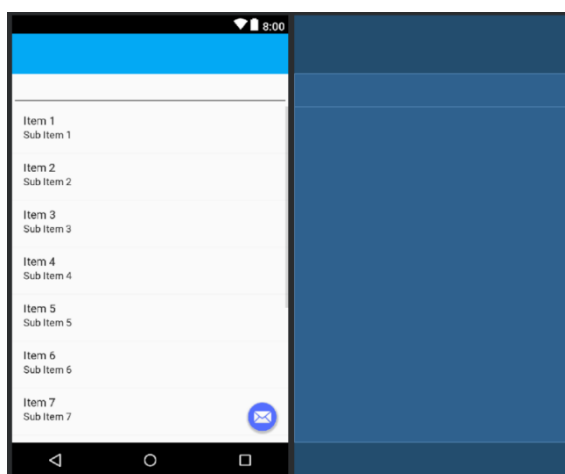


Figura 44. Interfaz de Consulta de Materiales.

Realizado por: Fabricio Maldonado.

La estructura de la interfaz es:

Componentes utilizados en Consulta de laboratorios.

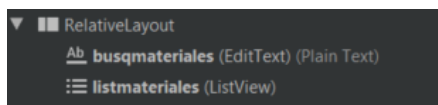


Figura 45. Componentes utilizados en ver Laboratorios.

Realizado por: Fabricio Maldonado.

Para crear la búsqueda dinámica se estructuró a un EditText que recorre el servicio web y lo agrega en un ListView.

Código búsqueda de Materiales desde un ListView.

```
adapter=new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,titulo);
listView.setAdapter(adapter);

buscador.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
        adapter.getFilter().filter(charSequence); }
})
```

Figura 46. Código búsqueda de Materiales desde un ListView.

Realizado por: Fabricio Maldonado.

3.2.5. Construcción de módulo mantenimiento.

El módulo de gestión de mantenimiento guarda un incidente de alguna maquina o máquinas de un laboratorio. La información es ingresada por un usuario.

El técnico personal de CECASIS, revisará el incidente, cambiará estado y escribirá el arreglo por medio de la aplicación móvil.

Interfaz de Ingreso de Incidente.

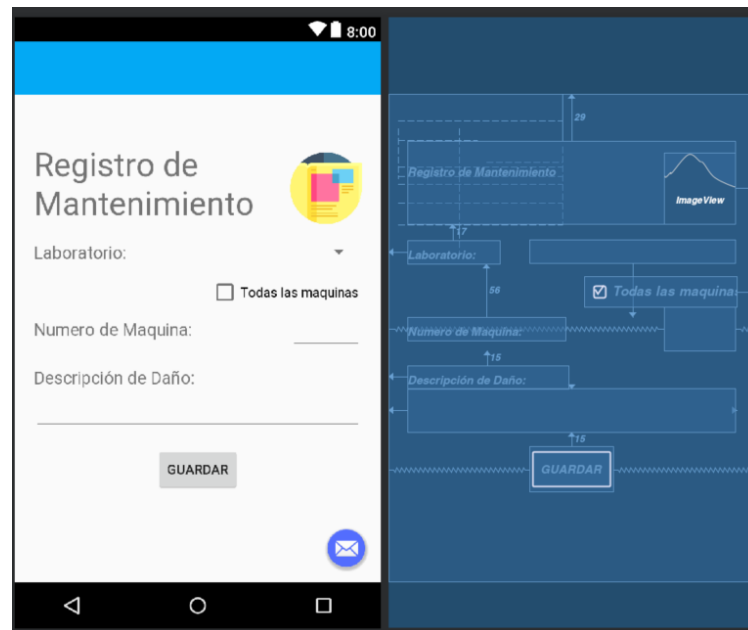


Figura 47. Interfaz de Ingreso de Incidente.

Realizado por: Fabricio Maldonado.

La estructura de la interfaz es:

Componentes utilizados en el ingreso de incidentes.

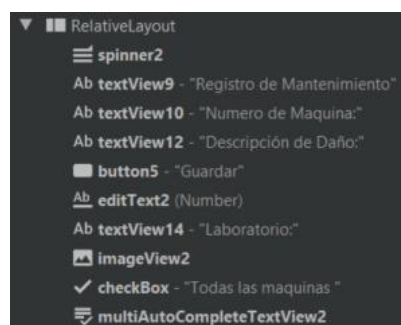


Figura 48. Componentes utilizados en el ingreso de incidentes.

Realizado por: Fabricio Maldonado.

La notificación de ingreso de nuevos incidentes se realiza mediante AlertDialog el cual ayuda a enviar la notificación al usuario indicado.

Código de AlertDialog para guardar el incidente.

```
AlertDialog.Builder mBuilder = new AlertDialog.Builder(MantenimientoAsignacion.this);
// mBuilder.setTitle("Noticias");

View mView =
getLayoutInflater().inflate(R.layout.activity_mantenimiento_asignacionpop,null);

mBuilder.setView(mView);
mBuilder.setNegativeButton("Guardar", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialogInterfaz, int i) {
guardarincidente();
}
});

AlertDialog mDialog = mBuilder.create();
mDialog.show();
```

Figura 49. Código de AlertDialog para guardar el incidente.

Realizado por: Fabricio Maldonado.

3.3. Construcción de servicios web.

Los servicios web más importantes en la aplicación web son:

3.3.1. Servicio web de autenticación e ingreso.

La dirección para consumir el servicio es:

Dirección: <http://localhost/serv/inicio.php>.

Los datos necesarios para consumir el servicio son:

Datos enviados y respuestas json de Autenticación e ingreso

Request

```
{ "usu": "fmaldonadota@est.ups.edu.ec",
  "pas": "1725269151",
  "claveapp": "8e2c7cb14cb0f429" }
```

Los datos que nos renvía el servicio son:

Response 200

```
[
{ "nombre_rol": "ESTUDIANTE",
  "nombre_usuario": "Fabricio",
  "apellido_usuario": "Maldonado",
  "cedula_usuario": "1725269151",
```

```

"key":"fe121f42e00308283268aff9d9cb0f2d8d17135b7313c467fe
35e75afabbf13f4bf03d80a5a2
1416f52429962689e2cd004a68e539c3c216ccab9fdde66e1b
ab"}
]

```

Figura 50. Datos enviados y respuestas json de Autenticación e ingreso.

Realizado por Fabricio Maldonado.

3.3.2. Servicio web de consulta de laboratorios.

La dirección para consumir el servicio es:

Dirección : <http://localhost/serv/labfiltro.php>.

Los datos necesarios para consumir el servicio son:

Datos enviados y respuestas json de consulta de laboratorios.

Request

```

{ "fecha": "29/07/2018 ",
  "hora": "17:00",
  "nummaquina": "8e2c7cb14cb0f429",
  "token":
  "fe121f42e00308283268aff9d9cb0f2d8d17135b7313c467fe35e75a
fabbf13f4bf03d80a5a2
1416f52429962689e2cd004a68e539c3c216ccab9fdde66e1b
ab"}

```

Los datos que nos renvía el servicio son:

Response 200

```

[
{"Laboratorio":"Laboratorio 10",
"numero de máquinas":"27"}
{"Laboratorio":"Laboratorio 7",
"numero de máquinas":"27"}
]

```

Figura 51. Datos enviados y respuestas json de consulta de laboratorios.

Realizado por Fabricio Maldonado.

Son varios datos ya que pueden ser varios laboratorios disponibles.

3.3.3. Servicio web de consulta de materiales.

La dirección para consumir el servicio es:

Dirección : <http://localhost/serv/materiales.php>.

Los datos necesarios para consumir el servicio son:

Datos enviados y respuestas json de consulta de materiales.

Request

```
{ "token":  
  "fe121f42e00308283268aff9d9cb0f2d8d17135b7313c467fe35e75a  
  fabbf13f4bf03d80a5a2  
    1416f52429962689e2cd004a68e539c3c216ccab9fdde66e1b  
  ab"}}
```

Los datos que nos renvía el servicio son:

Response 200

```
[  
  {  
    "Material": "Cautin",  
    "descripción": "Material eléctrico "  
    "disponibilidad ": "2"}  
  {  
    "Laboratorio": "Calculadora Casio gd44",  
    "descripción": "Material "  
    "disponibilidad ": "1"}  
]
```

Figura 52. Datos enviados y respuestas json de consulta de materiales.

Realizado por Fabricio Maldonado.

Son varios datos ya que pueden ser varios los materiales disponibles.

3.3.4. Servicio web de consulta de incidentes.

La dirección para consumir el servicio es:

Dirección : <http://localhost/serv/verIncidente.php>.

Los datos necesarios para consumir el servicio son:

Datos enviados y respuestas json de consulta de incidentes.

Request

```
{ "token":  
  "fe121f42e00308283268aff9d9cb0f2d8d17135b7313c467fe35e75a  
  fabbf13f4bf03d80a5a2  
    1416f52429962689e2cd004a68e539c3c216ccab9fdde66e1b  
  ab"} }
```

Los datos que nos renvía el servicio son:

Response 200

```
[  
  {  
    "NumeroIncidente": "23",  
    "descripción": "No funciona la maquina",  
    "Fecha": "29/07/2018",  
    "Laboratorio": "2",  
    "Maquina": "2",  
    "Asignación": "Tecnico Darwin"}  
].
```

Figura 53. Datos enviados y respuestas json de consulta de incidentes.

Realizado por Fabricio Maldonado.

4. Pruebas del software

4.1. Software y hardware.

Se realizó la prueba en:

Software:

- Windows Server 2016 Standard Evaluation (64 bits).

Hardware:

- Procesador: Common KVM porcessor 2.67 GHZ (2 procesadores).
- Ram: 9,77 GB.

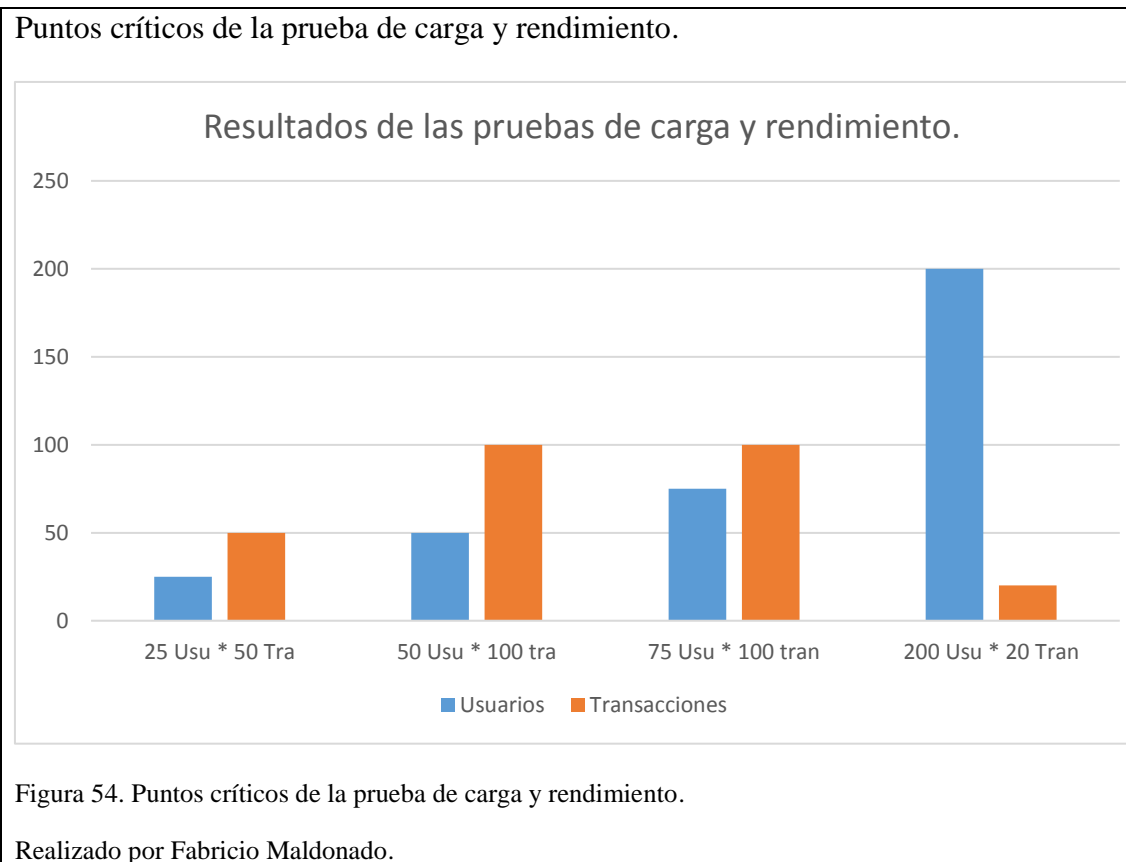
4.2. Prueba de rendimiento de la base de datos.

Número de Usuarios (Pool).	Número de Transacción por (usuario).	Número de Transacciones.	Número de datos en la tabla.	Número de campos en la Tabla.	Petición JDBC.	Transacciones Completas .	Rendimiento.	%Error.	Kb / seg.	Seg,
25	10	250 / s	30000	37	10	10 x usr	0.41 /seg	0,00 %	4848.11	14 seg
25	20	500 / s	30000	37	20	20 x usr	0.47 / seg	0.00 %	5536.72	25 seg
25	35	875 / s	30000	37	35	35 x urs	1.2 / seg	0.00 %	6345.56	28 seg
25	50	1250 / s	30000	37	50	50 x urs	1.7 / seg	50.00 %	6038.47	29 seg
25	100	2500/ s	30000	37	25	25 x urs	3.6 / seg	75.00 %	6319.42	27 seg
50	10	500 / s	30000	37	10	10 x urs	0.352 / seg	0.00 %	4076.68	17 seg
50	20	1000/ s	30000	37	20	20x urs	0.455 / seg	0.00 %	5259.19	26 seg
50	35	1750 / s	30000	37	35	35 x urs	0.438 / seg	0.00 %	5063.77	48 seg
50	50	2500/ s	30000	37	50	50 x urs	0.547 / seg	0.00 %	6321.78	55 seg
50	100	5000 / s	30000	37	50	50 x urs	1.6 / seg	50.00 %	5610.48	1:02 seg
75	10	750 / s	30000	37	10	10 x urs	0.27 / seg	0.00 %	3118.65	22 seg

Número de Usuarios (Pool).	Número de Transacción por (usuario).	Número de Transacciones.	Número de datos en la tabla.	Número de campos en la Tabla.	Petición JDBC.	Transacciones Completas .	Rendimiento.	%Error.	Kb / seg.	Seg,
75	20	1500 / s	30000	37	20	20 x urs	0.398 / seg	0.00 %	4598.68	30 seg
75	35	2625 / s	30000	37	35	35 x urs	0.41/ seg	0.00 %	4749.53	51 seg
75	50	3750 / s	30000	37	50	50 x urs	0.51/ seg	0.00 %	5918.84	58 seg
75	100	7500 / s	30000	37	75	75 x urs	1.0 / seg	25.00 %	5420.97	36 seg
100	10	1000 / s	30000	37	10	10 x urs	0.23 / seg	0.00 %	2663.50	26 seg
100	20	2000 / s	30000	37	20	20 x urs	0.342 / seg	0.00 %	3959.02	35 seg
100	35	3500/ s	30000	37	35	35 x urs	0.42 / seg	0.00 %	4261.43	50 seg
200	20	4000 /s	30000	37	0	0 x urs	5 min	100 %	922.80	5. min

Tabla 6. Valores de la prueba de carga y rendimiento de la BDD.

Realizado por Fabricio Maldonado.



Estos son los puntos críticos es decir cuando mediante la transacción existen perdidas de información.

4.3. Pruebas de aceptación.

Las pruebas de aceptación fueron realizadas para verificar la funcionalidad de la aplicación móvil y comprobar que realiza lo descrito en las historias de usuario, cada historia de usuario se ejecutará y demostrará las funciones de los módulos.

La prueba de aceptación estará formada de los siguientes puntos:

1. Caso de pruebas.
2. Número de caso.
3. Número de historia de usuario.

4. Nombre de caso de prueba.
5. Descripción.
6. Condiciones de ejecución.
7. Entradas.
8. Resultado esperado.
9. Evaluación.

4.3.1. Perfil de usuarios.

1. Estudiante.
2. Profesor.
3. Administrador.
4. Técnico del CECASIS.

Características de los perfiles:

Perfiles de Usuario.

Participante.	Nivel de Experiencia con Android.	Perfil de usuario.	Interfaz Táctil.	Género.	Edad.
1	Estudiante.	Alto.	Sí.	Hombre.	23
2	Estudiante.	Medio.	Sí.	Mujer.	24
3	Estudiante.	Bajo.	Sí.	Hombre.	20
4	Profesor.	Alto.	Sí.	Hombre.	47
5	Administrador.	Alto.	Sí.	Mujer.	49
6	Técnico de CECASIS.	Alto.	Sí.	Hombre.	35

Tabla 7. Perfiles de Usuario.

Realizado por Fabricio Maldonado.

4.3.1.1. Prueba de aceptación de login

Pruebas de Aceptación : Login.	
Caso de Prueba :	Numero de Historia de Usuario : 1.
Nombre de Caso de Prueba : Login.	
Descripción: La aplicación utiliza servicios web para reconocer el usuario y contraseña de usuario para lograr una conexión.	
Condiciones de Ejecución: El usuario debe ingresar el correo y la contraseña de la Universidad Politécnica Salesiana, para ser consumida por un servicio Web el cual verifica en la base de datos.	
Entradas: Correo institucional. Contraseña Institucional.	
Resultado Esperado: Ingresar a la aplicación.	
Evaluación: Positiva.	

Tabla 8. Prueba de aceptación: login.

Realizado por: Fabricio Maldonado.

4.3.1.2. Pruebas de aceptación: visualizar noticias.

	Pruebas de Aceptación: Visualizar Noticias.	
	Caso de Prueba: 02.	Numero de Historia de Usuario: .
	Nombre de Caso de Prueba: Visualizar noticias.	

	Descripción: La aplicación visualizara las ultimas noticias ingresadas por el administrador, para tener una mejor comunicación.
	Condiciones de Ejecución: Tener una sesión iniciada.
	Entradas: Hilo del usuario y contraseña.
	Resultado Esperado: Visualizar las ultimas noticias ingresadas desde la interfaz WEB.
	Evaluación: Positiva.
	Tabla 9. Pruebas de Aceptación; Visualizar Noticias.
Realizado por: Fabricio Maldonado.	

4.3.1.3. Pruebas de aceptación: ver detalle de noticias.

Pruebas de Aceptación : Ver detalle de noticias.	
Caso de Prueba : 1.	Numero de Historia de Usuario : 2.
Nombre de Caso de Prueba : Ver detalle de noticias.	
Descripción: El módulo de noticias tiene un detalle el cual se lo puede ver dando un click en la noticia.	
Condiciones de Ejecución. Ingresar a una noticia de interés personal.	
Entradas: Interacción con el módulo de noticias.	
Resultado Esperado: Visualizar de manera amigable las noticias y su descripción.	
Evaluación: Positiva.	

Tabla 10. Pruebas de Aceptacion: Ver detalles de Noticias.

Realizado por: Fabricio Maldonado.

4.3.1.4. Pruebas de aceptación: login recordando contraseña.

Pruebas de Aceptación: Login recordando contraseña.	
Caso de Prueba : 1.	Numero de Historia de Usuario : 2.
Nombre de Caso de Prueba :Login recordando contraseña.	
Descripción: Ingresar al Módulo de noticias sin necesidad de un usuario y contraseña.	
Condiciones de Ejecución: Guardar previamente el Usuario y contraseña con Shared Preference en Android.	
Entradas: Usuario y contraseña en Shared preference.	
Resultado Esperado: Ingreso al módulo de noticias sin usuario y contraseña.	
Evaluación: Positiva.	

Tabla 11. Pruebas de Aceptación: Login recordando Contraseña.

Realizado por: Fabricio Maldonado.

4.3.1.5. Pruebas de aceptación: buscar y visualizar laboratorios disponibles.

Pruebas de Aceptación : Buscar y visualizar laboratorios disponibles.

	Caso de Prueba: 1.	Numero de Historia de Usuario : 3.
	Nombre de Caso de Prueba : Buscar y visualizar laboratorios Disponibles.	
	Descripción: Con las variables seleccionadas buscara el laboratorio de acuerdo a las necesidades del usuario.	
	Condiciones de Ejecución: Ingreso correcto de Fechas, horas, número de máquinas y software disponible.	
	Entradas: Variables correctamente ingresadas.	
	Resultado Esperado: Visualizar los laboratorios disponibles con las variables ingresadas.	
	Evaluación: Positiva.	

Tabla 12. Prueba de aceptación: Buscar y Visualizar Laboratorio disponibles.

Realizado por: Fabricio Maldonado.

4.3.1.6. Pruebas de aceptación: visualizar los materiales disponibles.

	Pruebas de Aceptación : Visualizar los materiales Disponibles.	
	Caso de Prueba :	Numero de Historia de Usuario: 3.
	Nombre de Caso de Prueba : Visualizar los materiales Disponibles.	
	Descripción: Visualizar los materiales disponibles en el cecasis.	
	Condiciones de Ejecución:	

Mediante servicio web obtener los materiales disponibles para su verificación.
Entradas: Ingreso al módulo extrayendo el usuario y contraseña de Shared preference.
Resultado Esperado: Visualizar los materiales disponibles.
Evaluación: Positiva.

Tabla 13. Prueba de Aceptación: Visualizar los materiales Disponibles.

Realizado por: Fabricio Maldonado.

4.3.1.7. Pruebas de aceptación: ingresar incidencia.

Pruebas de Aceptación: Ingresar Incidencia.	
Caso de Prueba : 1.	Numero de Historia de Usuario : 4.
Nombre de Caso de Prueba : Ingreso al módulo mantenimiento y registrar una incidencia.	
Descripción: Ingresar al módulo de registro de incidencias, y registrar la incidencia.	
Condiciones de Ejecución: Ingreso al módulo Mantenimiento, reconociendo en usuario y la contraseña.	
Entradas: Ingreso al módulo extrayendo el usuario y contraseña de Shared preference.	
Resultado Esperado: Ingreso y registrar sin error.	
Evaluación: Positiva.	

Tabla 14. Prueba de Aceptación: Ingresar Incidencia.

Realizado por Fabricio Maldonado.

4.3.1.8. Pruebas de aceptación: ver incidencias.

5. Pruebas de Aceptación : Ver incidencia registradas.	
Caso de Prueba 1.	Numero de Historia de Usuario 4.
Nombre de Caso de Prueba : Visualizar las incidencias reigstras.	
Descripción : Entrar al módulo de Mantenimiento y ver incidencias registradas.	
Condiciones de Ejecución : Ingreso al módulo Mantenimiento, reconociendo en usuario y la contraseña.	
Entradas: Usuario y contraseña del Shared preference junto al servicio web consumido.	
Resultado Esperado: Visualizar los valores adecuados según el usuario registrado.	
Evaluación: Positiva.	
<p>Tabla 15. Pruebas de Aceptación: Ver incidencias.</p> <p>Realizado por Fabricio Maldonado.</p>	

4.3.1.9. Pruebas de aceptación: ver asignación de incidencias.

Pruebas de Aceptación : Ver Asignación de incidencias.	
Caso de Prueba 1.	Numero de Historia de Usuario 4.
Nombre de Caso de Prueba : Asignación de Incidencia.	

	<p>Descripción:</p> <p>Ingresar al módulo de mantenimiento con el perfil de técnico y visualizar las incidencias asignadas al usuario.</p>
	<p>Condiciones de Ejecución : Usuario y contraseña extraído del Shared preference.</p>
	<p>Entradas: Usuario y tipo de Usuario.</p>
	<p>Resultado Esperado: Visualizar las incidencias de acuerdo al usuario logueado.</p>
	<p>Evaluación: Positiva.</p>
<p>Tabla 16. Prueba de Aceptación: Ver Asignación de Incidencias.</p> <p>Realizado por Fabricio Maldonado.</p>	

Conclusiones

- El levantamiento y optimización de los procesos con la notación BPMN del CECASIS proporciona de manera gráfica un estándar de procesos que es fácil entendible y legible para personas no expertas.
- El desarrollo de la aplicación móvil utilizando Scrum fue incremental interactiva y permitió obtener los requerimientos, análisis, diseño evolución y entrega.
- El número de usuarios conectados al servicio web por defecto son 100 con 75 transacciones por segundo.
- La utilización de servicios web es una solución efectiva, debido a que, con esta tecnología, se consigue evitar la duplicidad al contar con una sola conexión y optimizar recursos al trabajar con datos semiestructurados.

Recomendaciones

- Utilizar dispositivos móviles Android con sistemas operativos superiores al 4 JellyBean en adelante para garantizar el correcto funcionamiento de la aplicación móvil.
- Utilizar Apache 2 y PHP5.7 para la implementación de los servicios web ya que cuenta con componentes específicos que no tienen versiones anteriores.
- Utilizar estándares de Material Designer para que la aplicación móvil tenga mejor usabilidad para el usuario final.
- Cargar las dependencias de Gradle localmente para que el IDE se más fluido en su trabajo.

Bibliografía

- Apache Friends. (2018). *¿Qué es XAMPP?* Obtenido de <https://www.apachefriends.org/es/index.html>.
- Gradle Inc. (2018). *Gradle Release Notes*. Obtenido de <https://docs.gradle.org/current/release-notes.html>.
- Apache Software Foundation. (2018). *Apache JMeter*. Obtenido de <https://jmeter.apache.org/>.
- Apache Software Foundation. (2018). *What is the Apache HTTP Server Project?* Obtenido de http://httpd.apache.org/ABOUT_APACHE.html.
- Bizagi. (2018). *La Plataforma de Negocios Digitales*. Obtenido de <https://www.bizagi.com/es>.
- Debevoise, N. T. (2016). The MicroGuide to Process Modeling in BPMN. En N. T. Debevoise.
- developer.android.com. (09 de 2017). *Andorid Studio Notes* . Obtenido de <https://developer.android.com/studio/releases>.
- Evolus. (2018). *Pencil Proyect* . Obtenido de <https://pencil.evolus.vn/>.
- Google. (2018). *Material Designer para Android* . Obtenido de <https://developer.android.com/design/material/?hl=es-419>.
- Google. (2018). *Support Library Packages*. Obtenido de <https://developer.android.com/topic/libraries/support-library/packages>.
- Group, T. P. (s.f.). *Json_encode*. Obtenido de <http://php.net/manual/es/function.json-encode.php>.
- Maner, W. (2018). *PROTOTIPADO*. Obtenido de <https://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>
- Nederlands, D. (2018). *Introducing JSON*. Obtenido de <https://www.json.org/>.
- Oracle. (2018). *Java Development*. Obtenido de <http://www.oracle.com/technetwork/java/javase/10-0-1-relnotes-4308875.html>.
- Quilumba, M. (2018). *AUTOMATIZACIÓN DEL PROCESO DE RESERVACIÓN DE LABORATORIOS, PRÉSTAMOS Y MANTENIMIENTO DE EQUIPOS TECNOLÓGICOS DEL CECASIS, A TRAVÉS DE UNA APLICACIÓN WEB*.
- Riggs, S. (2018). *Postgresq*. Obtenido de <https://www.postgresql.org/community/>.
- Smith, J. (2016). *Android Asynchronous Http Client*. Obtenido de <http://loopj.com/android-async-http/>.
- Smith, J. (2016). *Android Smart Image View*. Obtenido de <http://loopj.com/android-smart-image-view/>.

- The PHP Group. (2018). *¿Qué es PHP?* Obtenido de <http://php.net/manual/es/intro-what-is.php>.
- The PHP Group. (2018). *Objetos de datos de PHP* . Obtenido de <http://php.net/manual/es/book.pdo.php>.
- Wang, X. Y. (2016). PowerDesigner Chief Architect,. En X. Y. Wang.
- World Wide Web. (2016). *RESTful Web Services - Introduction*. Obtenido de https://www.tutorialspoint.com/restful/restful_environment.htm.